

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Regina Padaigienė

**LENKIAMŲ DAUGIASLUOKSNIŲ KONSTRUKCINIŲ
ELEMENTŲ PROJEKTAVIMO PROCESO MODELIAVIMAS**

Magistro darbas

Darbo vadovas:

Doc., dr. S.Maciulevičius

Kaunas
2004

TURINYS

1. ĮVADAS	2
2. DAUGIASLUOKSNIŲ KONSTRUKCINIŲ ELEMENTŲ BENDROSIOS CHARAKTERISTIKOS.....	4
2.1. Neutraliosios ašies koordinatės nustatymas.....	5
2.2. Konstrukcinio elemento ekvivalentinis tamprumo modulis	5
2.3. Sijos standumas lenkimo atveju.....	6
2.4. Normalinių įtempimų nustatymas.....	6
2.5. Tangentinių įtempimų apskaičiavimas	7
2.6. Konstrukcijos stiprumo sąlygos.....	7
2.7. Modeliavimo programą sudarančių dalių aprašymas	8
3. PROJEKTINĖ DALIS	10
3.1 Reikalavimų projektuojamai sistemai specifikacija	10
3.1.1 Reikalavimai funkcionavimui.....	10
3.1.2 Reikalavimai vartotojo sąsajai	12
3.1.3 Eksploatavimo aplinka.....	12
3.1.4 Apribojimai	13
3.1.5 Duomenų šrantai	13
3.2 Duomenų struktūra	14
3.3 Projektuojamos sistemos architektūra	14
3.4. Programinių modulių specifikacijos	17
3.4.1 Modulių pavadinimai	17
3.4.2 Modulių funkciniai aprašymai	17
3.4.3 Modulio duomenų struktūra.....	19
3.4.4 Modulio sąsaja, pasirinkta dialogo struktūra	20
3.5 Testavimas	21
4. VARTOTOJO DOKUMENTACIJA.....	26
4.1. Funkcinis sistemos aprašymas	26
4.2. Sistemos vadovas.....	26
4.3. Sistemos instaliavimo dokumentas.....	34
PRODUKTO KOKYBĖS ĮVERTINIMAS.....	35
PAGRINDINIAI REZULTATAI IR IŠVADOS.....	36
LITERATŪRA:	37
SUMMARY.....	38
PRIEDAI.....	39

LENKIAMŲ DAUGIASLUOKSNIŲ KONSTRUKCINIŲ ELEMENTŲ PROJEKTAVIMO PROCESO MODELIAVIMAS

1. ĮVADAS

Daugelis konstrukcinių elementų gaminami iš vienos rūšies medžiagos, todėl daugeliu atveju neįmanoma suderinti medžiagos stiprumo savybių su konstrukcijai keliamais masės, kainos ir kitais reikalavimais.

Jau prieš daugelį metų žmogus pastebėjo, kad derinant kai kurias natūralias medžiagas (molį, medį, akmenį ir pan.) galima pasiekti tokių dirbinių savybių, kurių neturi pavienė medžiaga. Nuolat atsirandančios naujos medžiagos sudaro galimybes priimti naujus techninius sprendimus ir taikyti naujas gamybos technologijas. Dauguma šiuolaikinių konstrukcinių medžiagų gaminamos naudojant įvairias skirtingas medžiagas. Daugiasluoksnės konstrukcijos gali būti sudarytos iš skirtingų medžiagų: plastmasių, kompozitų, metalų ir t.t.

Lenkiamą tiesų strypą įprasta vadinti sija. Dažnai mechaninėse sistemose sunku arba visai neįmanoma išsiversti be tokių konstrukcinių elementų kaip sijos. Konstrukcinius elementus galima laikyti sudėtingo skerspjūvio sijomis. Norint gauti optimalią konstrukciją, būtina parinkti sluoksnių geometriją ir medžiagą.

Nors literatūroje yra pateikta įvairių daugiasluoksnių konstrukcijų elementų skaičiavimo metodikų, tačiau daugeliu atveju jos matematiškai yra gana sudėtingos ir sunkiai įsisavinamos. Viena iš lengvai suprantamų ir lengvai įsisavinamų yra mokomoji knyga J.Bareišio, V.Paulausko „Daugiasluoksnių kompozicinių konstrukcinių elementų projektavimas“. Kol kas moksliniams tyrimams, inžineriniams skaičiavimams, mokymo tikslams ši skaičiavimo metodika mažai taikoma, nes dauguma skaičiavimų atliekama rankiniu būdu, sugaištama daug laiko, bei galimi apsirikimai įvedant duomenis. Todėl, natūralu, kad šioje vietoje atsiranda poreikis turėti paprastesę ir patogesnę skaičiavimo priemonę. Dėl šios priežasties pasirinkau temą „Lenkiamų daugiasluoksnių konstrukcinių elementų projektavimo proceso modeliavimas“. Šio darbo tikslas yra sudaryti paprastą ir nesudėtingą programą, konkrečiai - lenkiamų daugiasluoksnių konstrukcinių elementų projektavimo proceso modeliavimo sistemą, kuri apimtų pagrindinių parametų skaičiavimus bei pateiktų grafinius vaizdus.

Šio darbo pagrindinė idėja yra parengti lenkiamų daugiasluoksnių konstrukcinių elementų projektavimo proceso modelį, kuris įgalintų:

- pagal pasirenkamus pradinius duomenis nubraižyti sijos skerspjūvio formą;

- apskaičiuoti pagrindinius konstrukcijos stiprumo ir standumo parametrus (įtempimus, standumą D ir kt.);
- pagal apskaičiuotų įtempimų reikšmes nubrėžti jų pasiskirstymą daugiasluoksnių konstrukcinių elementų skerspjūvyje.

Šiam projektui realizuoti naudoju *Visual Basic* [3] programinę įrangą, nes ji yra gana universali ir tinka įvairiems projektavimo uždaviniams spręsti. Žemiau darbe yra pateiktas detalesnis šio modelio aprašymas.

Programinei įrangai buvo paruoštas projektas:

- atlikta išsami esamos programinės įrangos analizė, parinkti sprendimų realizavimo keliai;
- išanalizuoti vartotojo reikalavimai bei išskirti funkciniai reikalavimai;
- sudarytas projektuojamos sistemos architektūrinis modelis;
- išskirti duomenų šrantai ir galimos duomenų struktūros;
- nustatyti apribojimai ir eksploataavimo aplinka;
- sukurta vartotojo sąsaja;
- atliktas sistemos testavimas;
- parengta vartotojo dokumentacija, susidedanti iš tokių dalių:
 - sistemos funkcinio aprašymo;
 - detalaus sistemos aprašymo;
 - sistemos įdiegimo instrukcijos.

Darbe atlikta išsami esamos programinės įrangos analizė, sudarytas lenkiamų daugiasluoksnių konstrukcinių elementų projektavimo proceso modelis. Ši programa skirta daugiasluoksnių konstrukcijų, kurios yra veikiamos skersinių jėgų ir lenkimo momentų, standumo ir stiprumo skaičiavimui.

2. DAUGIASLUOKSNIŲ KONSTRUKCINIŲ ELEMENTŲ BENDROSIOS CHARAKTERISTIKOS

Pastaruoju metu vis daugiau šiuolaikinių konstrukcijų yra sudarytos iš kelių naujausių medžiagų, kurios konstrukcijai suteikia būtinas technines, eksploatacines ir kitas savybes. Bendro įvairių medžiagų darbo rezultatas yra tolygus naujos medžiagos, kurios savybės kokybiškai ir kiekybiškai skiriasi nuo kiekvieno komponento savybių, darbo rezultatams. Parinkus kompozitines medžiagas su skirtingomis mechaninėmis ir fizikinėmis savybėmis, galima sukurti optimalių parametrų konstrukcinius elementus, pasižyminčius didžiausiu stiprumu ir standumu bei mažiausiu tankiu ar kaina.

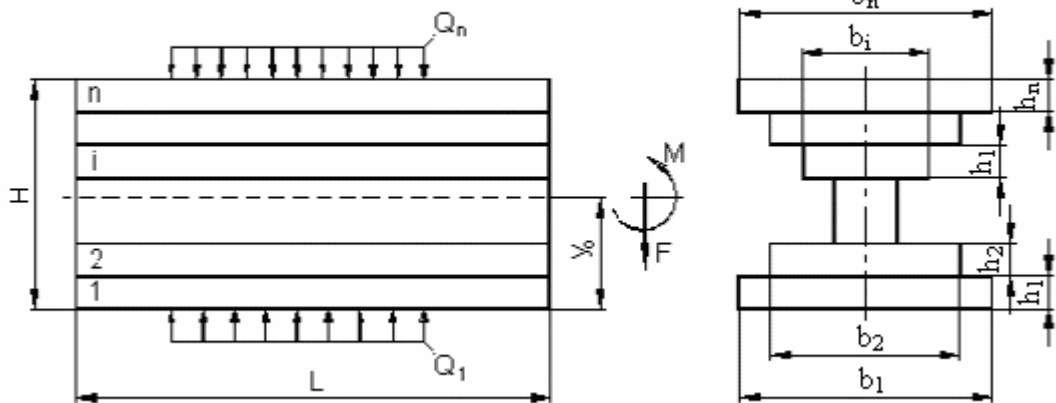
Daugiasluoksnės konstrukcijos plačiai naudojamos įvairiose srityse, pradedant buitine technika ir baigiant įvairios paskirties skraidymo aparatais. Konstrukciniai elementai, pagaminti iš šių medžiagų, gali būti veikiami statinių ir ciklinių apkrovų, išorinės aplinkos faktorių. Minėtiems poveikiams veikiant gali įvykti nepageidaujamas medžiagos fizinių bei mechaninių savybių pasikeitimas. Todėl labai svarbu mokėti gerai apskaičiuoti daugiasluoksnių konstrukcijų, sudarytų iš medžiagų su skirtingomis fizinėmis ir mechaninėmis charakteristikomis, stiprumo, standumo, stabilumo rodiklius, nusakančius gaminio saugaus eksploatavimo sąlygas.

Šių uždavinių realizavimui būtina turėti analitines išraiškas ir mokėti jomis tinkamai pasinaudoti. Todėl būtina teisingai nustatyti:

- neutraliojo sluoksnio koordinatę y_0 ;
- konstrukcinio elemento ekvivalentinį tamprumo modulį E_k ;
- įvairių veiksnių įtaką daugiasluoksnių konstrukcijų elementų standumui D ;
- normalinius įtempimus σ_j bet kuriame sijos sluoksnyje bei tangentinius įtempimus

τ_j .

Bendruoju atveju sluoksniuotą konstrukcinį elementą gali sudaryti įvairaus pločio b ir aukščio (storio) h sluoksniai (1 pav.). Konstrukcinis elementas gali būti apkrautas įvairių tipų apkrovomis: ašine jėga N , skersine jėga F , sloginiu Q ar lenkimo momentu M arba kombinuota apkrova. Kiekviename sijos skerspjūvyje bendru atveju veikia ašinės ir skersinės jėgos bei lenkimo momentų įrašos.



1 pav. Sluoksniuotas konstrukcinis elementas

2.1. Neutraliosios ašies koordinatės nustatymas

Sijos ašis, kurios ilgis lenkimo metu nekinta, vadinamas neutraliąja ašimi. Jos koordinatė nuo daugiasluoksnio konstrukcinio elemento apatinio ar viršutinio sluoksnio pagal [1] apskaičiuojama pagal formulę:

$$y_0 = \frac{\sum_{j=1}^n B_j h_j + 2 \sum_{j=1}^n B_j \cdot \sum_{k=1}^{j-1} h_k}{2 \sum_{j=1}^n B_j} \quad [\text{m}],$$

čia B - ašinis standumas $B = \sum E_i A_i$ [N];

h_j - j -tojo sluoksnio storis [m];

E_j - j -tojo sluoksnio tamprumo modulis [Pa].

Todėl norint rasti neutraliosios ašies padėtį daugiasluoksniėje sijoje pakanka žinoti kiekvieno sluoksnio medžiagos tamprumo modulį ir geometrinius matmenis.

2.2 Konstrukcinio elemento ekvivalentinis tamprumo modulis

Daugiasluoksnės sijos ekvivalentinis tamprumo modulis lenkimo atveju yra atskirų jį sudarančių elementų tamprumo modulių, padaugintų iš šio elemento inercijos momento santykio su visos sijos skerspjūvio inercijos momentu, suma. Jis gali būti paskaičiuotas pagal formulę [1]:

$$E_k = \sum_{j=1}^n E_j I_j / I_k,$$

čia I_j - j -tojo sluoksnio skerspjūvio ploto inercijos momentas neutraliosios ašies atžvilgiu [m^4];

I_k – skerspjūvio inercijos momentas neutralios ašies atžvilgiu.

2.3 Sijos standumas lenkimo atveju

Izotropinių (kurių savybės vienodos visomis kryptimis) sijų tamprumo modulio ir inercijos momento sandauga $E_k I_k$ vadinama sijos standumu lenkimo atveju. Daugiasluoksnių sijų standumas apskaičiuojamas įvertinant siją sudarančių sluoksnių skaičių n , kurių kiekvienas turi savąjį standumą $E_j I_j$. Todėl pagal [1]:

$$D = E_k I_k = \sum_{j=1}^n E_j I_j \quad [\text{Nm}^2].$$

čia E_k – tamprumo modulis;

I_k – inercijos momentas neutralios ašies atžvilgiu.

Daugiasluoksnės sijos standumui nustatyti pakanka apskaičiuoti kiekvieno sluoksnio inercijos momentus sijos skerspjūvio neutralios ašies atžvilgiu ir rasti jų sandaugų su tamprumo moduliais sumą.

2.4 Normalinių įtempimų nustatymas

Normalinių įtempimų apskaičiavimų formulė [1] buvo gauta grynojo lenkimo atvejui (kai nėra skersinės jėgos). Laikoma, kad sijos, deformuojamos tamprumo ribose, įtempimai pasiskirsto pagal laužytą kreivę.

Apskaičiuojant daugiasluoksnės sijas buvo laikomasi šių prielaidų:

1. Besideformuojančios sijos skersiniai pjūviai lieka plokšti ir statmeni išilginiams sijos sluoksniams nepriklausomai nuo to, ar sijos medžiaga linijiškai tampri.

2. Sluoksnių sujungimo vietose nėra praslydimo.

3. Sluoksnio mechaninės charakteristikos priklauso nuo kompozito armuojančios ir rišamosios medžiagos, kaip visumos, savybių, nustatomų eksperimentiniu būdu.

$$\sigma_j = \frac{M \cdot y_j}{D} \cdot E_j \text{ [Pa];}$$

čia M - lenkimo momentas [Nm];

y_j - atstumas nuo neutraliosios ašies iki nagrinėjamojo sluoksnio [m].

Pereinant iš vieno sijos sluoksnio į kitą visada gaunamas įtempimų šuolis, proporcingas tų sluoksnių tamprumo modulių santykiui. Normalinių įtempimų daugiasluoksnėse sijose dydis priklauso ne tik nuo išorinės apkrovos, bet ir nuo sijos standumo bei medžiagos tamprumo modulio skaičiuojamame pjūvyje.

2.5 Tangentinių įtempimų apskaičiavimas

Kai lenkiamos sijos skerspjūvį veikia ne tik lenkimo momentas, bet ir skersinė jėga (kai lenkimas nėra grynasis), įvairiuose skerspjūvio taškuose veikia ne tik normaliniai, bet ir tangentiniai įtempimai, kurie apskaičiuojami pagal formulę [1]:

$$\tau_j = \frac{F \cdot C}{D \cdot b_j} \text{ [Pa];}$$

čia F – skersinė jėga [N];

b_j - sijos sluoksnio plotis [m];

C - ekvivalentinis statinis momentas neutralios ašies atžvilgiu [m³].

2.6 Konstrukcijos stiprumo sąlygos

Konstrukcijos stiprumas vertinamas pagal energetinę stiprumo teoriją [2].

$$\sigma_{ij} = \sqrt{\sigma_j^2 + 3\tau_j^2} \leq \sigma_{adm_j};$$

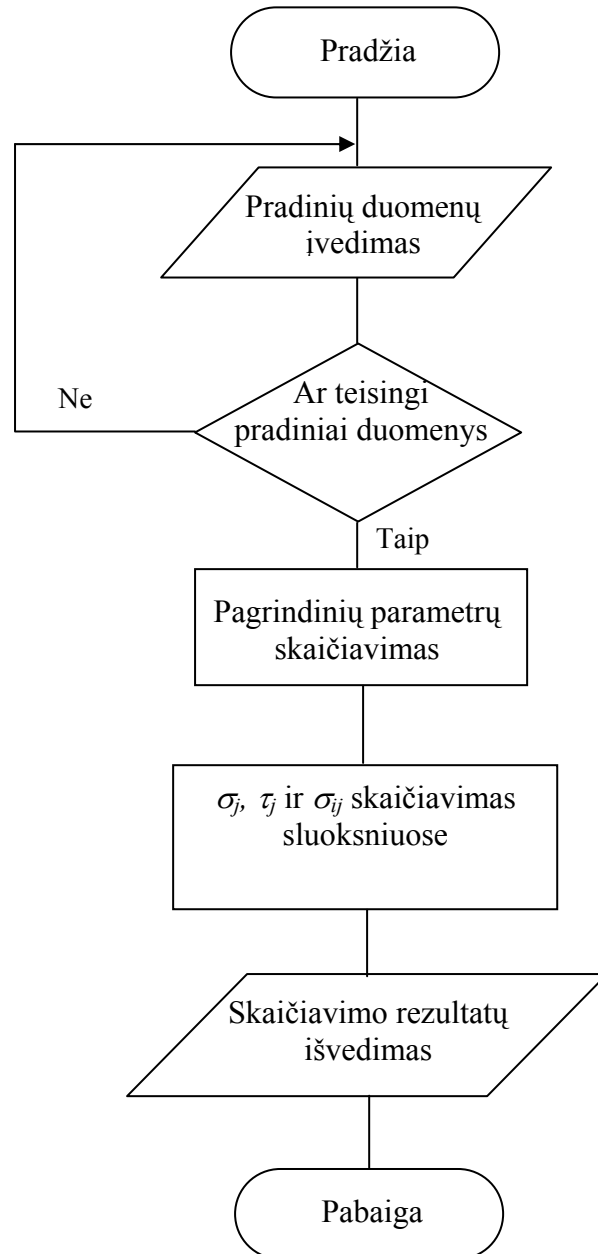
čia σ_{adm_j} – leistinasis sluoksnio medžiagos įtempimas [Pa];

σ_{ij} – įtempimų intensyvumas j -ame sluoksnyje [Pa].

Konstrukcija, netenkinanti šių sąlygų įgaus liekamąsias deformacijas. Atsiranda prielaida, kad ji nedarbinga ir neatlieka visos savo funkcijos. Optimali konstrukcija, kai kiekvieno sluoksnio *max* įtempimų intensyvumas yra artimas leistiniems įtempimams.

2.7 Modeliavimo programą sudarančių dalių aprašymas

Tokiu būdu pateikti duomenys yra prielaida proceso modeliavimo algoritmo sudarymui. Šio algoritmo struktūrinė schema pateikta žemiau (2 pav.).



2 pav. Skaičiavimo algoritmas

Lenkiamų daugiasluoksnių konstrukcinių elementų projektavimo proceso modeliavimo programą sudarytų šios dalys: pradinių duomenų įvedimas, pagrindinių parametrų skaičiavimas ir τ_j (tangentinių įtempimų), σ_j (normalinių įtempimų) bei σ_{ij} (įtempimų intensyvumo) skaičiavimas sluoksniuose.

Pradiniai skaičiavimo duomenys yra:

- lenkiamos sijos skerspjūvį veikiančios apkrovos:
 - lenkimo momentas M ;
 - skersinė jėga F ;
- lenkiamą siją sudarančių skirtingų medžiagų sluoksnių skaičius n ;
- lenkiamos sijos parametrai:
 - sluoksnio plotis b_j ;
 - aukštis h_j ;
- lenkiamos sijos sluoksnio medžiagos tamprumo modulis E_j .

Pagrindiniai skaičiavimo parametrai yra:

- j -ojo sluoksnio skerspjūvio plotas A_j ;
- j -ojo sluoksnio standumas tempimui B_j ;
- neutraliojo sluoksnio padėtis y_0 ;
- j -ojo sluoksnio skerspjūvio inercijos momentas I_j ;
- daugiasluoksnės sijos standumas lenkimui D .

Įtempimų paskaičiavimas sluoksniuose:

- tangentinių τ_j ;
- normalinių σ_j ;
- intensyvumo σ_{ij} .

Daugiasluoksnių konstrukcinių elementų panaudojimas yra efektyvus būdas siekiant pagerinti gaminio technines charakteristikas, o pateiktoji metodika – puikus įrankis juos naudoti praktikoje.

Aprašyto proceso modelis yra nesudėtingas, lengvai suprantamas kiekvienam mechanikos srities studentui, inžinieriui. Lenkiamų daugiasluoksnių konstrukcijų elementų projektavimo proceso modeliavimas atliekamas mokymosi tikslams. Visi reikalingi resursai ir priemonės vartotojams yra prieinami nemokamai.

Programoje keičiant keisti įvairius parametrus galima modeliuoti įvairias sijų konstrukcijas. Vartotojas iš įvairių konstrukcijų variantų turėtų išsirinkti jo manymu, geriausią. Šiam konstrukcinių elementų modeliavimui alternatyva būtų skaičiavimas naudojant tradicines priemones (skaičiuotuvą, logaritminę liniuotę ir pan.).

3. PROJEKTINĖ DALIS

3.1 Reikalavimų projektuojamai sistemai specifikacija

Projektuojama programa gali būti taikoma tiek moksliniams tyrimams, tiek inžineriniams skaičiavimams, tiek mokymo tikslams. Siūlomą modelį galės naudoti: dėstytojai - aiškindami studentų mokymo procese apie „Daugiasluoksnių kompozicinių konstrukcinių elementų projektavimą“; studentai - šią programą taikys uždavinių sprendimuose, laboratorinių bei kursinių projektų skaičiavimuose.

Pagrindiniai **projekto tikslai** yra šie:

- sukurti nesunkiai įsisavinamą, nebrangią programinę įrangą, leidžiančią vartotojui projektuoti lenkiamų daugiasluoksnių konstrukcijų elementus;
- sukurti vartotojui patogią sistemą, pagal pateiktus duomenis grafiškai atvaizduojančią konstrukciją;
- sukurti konstrukcijos charakteristikų priklausomybę nuo keičiamų parametrų.
- sudaryti lengvai valdomą programos modelį, kuris įgalintų vartotoją greitai ir kokybiškai apskaičiuoti bei modifikuoti konstrukcijos sluoksnius.

Projektui **keliami uždaviniai** yra šie:

- išsiaiškinti vartotojo norus ir pageidavimus programinei įrangai;
- išsiaiškinti daugiasluoksnių konstrukcijų skaičiavimo ypatumus;
- išsiaiškinti pasirenkamos grafinės sistemos galimybes;
- apibrėžti duomenų struktūrų modelį;
- apibrėžti vartotojo sąsają;
- nustatyti sistemos testavimą.

3.1.1 Reikalavimai funkcionavimui

Programinė įranga yra skirta jos vartotojui. Vartotojais gali būti: inžinieriai, dėstytojai, studentai bei asmenys atliekantys konstrukcijų mechaninius skaičiavimus. Todėl pirmiausiai išsiaiškinau jų poreikius.

Vartotojo poreikiai kuriamai programinei įrangai:

Vartotojas nori programos, kuri leistų suprojektuoti daugiasluoksnes kompozicines konstrukcijas:

- peržiūrėti suvestus duomenis bei juos koreguoti ar papildyti;

- atlikti skaičiavimus;
- pateikti grafinį sijos skersmens vaizdą;
- grafiškai atvaizduoti skaičiavimų rezultatus;
- gautus rezultatus išvesti į ekraną;
- gautus rezultatus išsaugoti faile;
- gautus rezultatus atspausdinti;
- turėti galimybę įvesti naujas sijos medžiagas arba pašalinti jau esamas;
- keisti sluoksnių geometrinius parametrus;
- keisti sluoksnius vietomis.

Daugiasluoksnių sijų lenkimo elementų projektavimas:

- Pradinių duomenų parinkimas;
- Skaičiavimo rezultatų išvedimas;
- Grafinis kiekvieno sluoksnio įtempimų vaizdavimas.

Keliami reikalavimai daugiasluoksnių lenkiamų sijų projektavimo sistemai:

- įvesti duomenis apie apkrovas veikiančias sijos skersmenyje;
- įrašyti arba pasirinkti sijos medžiagas;
- pridėti arba pašalinti medžiagą;
- įvesti sijos skerspjūvio geometrinius parametrus;
- pasirinkti sluoksnių skaičių;
- pridėti arba pašalinti sluoksnį;
- sukeisti vietomis sluoksnius;
- pagal pateiktus duomenis pavaizduoti sijos grafinį sluoksnių skerspjūvį;
- atlikti skaičiavimus;
- atvaizduoti skaičiavimų rezultatus;
- grafiškai pateikti normalinių, tangentinių įtempimų reikšmes bei įtempimų intensyvumą kiekvienam sluoksniui;
- leisti išsaugoti bei atsispausdinti gautus rezultatus.

Pagrindiniai reikalavimai patikimumui ir kokybei:

- programa turi būtų paprasta naudojimuisi;
- turi skaičiuoti be klaidų;
- turi aiškiai pateikti rezultatus;
- grafiniai vaizdai turi atitikti standartus;
- grafiniuose įtempimų epiūrose turi būti pateikiama detali informacija.

3.1.2 Reikalavimai vartotojo sąsajai

- Vartotojo sąsaja turi būti kuo paprastesnė ir suprantamesnė, lengvai valdoma.
- Ji turi būti taip sukomponuota, kad vartotojas kaip galima per trumpesnį laiką galėtų išmokti naudotis programa ir pradėtų dirbti.
- Turi būti neperkrauta nereikalingais elementais, o esami elementai aiškiai ir patogiai išdėstyti.
- Formų ir joje esančių elementų spalvos neturi būti per daug skirtingos.
- Projektuojamą vartotojo grafinę sąsają, turi sudaryti:
 - pagrindinės formos;
 - pagalbinės formos;
 - tekstiniai laukai;
 - valdymo elementai (mygtukai, prasukimo juostos);
 - vėliavėlės;
 - antraštės.
- Programos meniu turi būti viršutinėje pagrindinių formų dalyje.
- Komandų pasirinkimas turi būti patogus: meniu, įrankiais bei klavišais.
- Keleto failų langų peržiūra vienu metu.
- Medžiagų atsparumo projektuotojams būdingų terminų naudojimas.
- Aiškūs pranešimų dialogo langai, leidžiantys gauti papildomą informaciją.
- Vartotojui turi būti prieinama trumpa ir aiški pagalba.
- Paprastas klaidų apdorojimas ir kad būtų lengva atšaukti veiksmus.
- Išnaudoti pilną ekrano plotį.
- Naudoti didžiąsias bei mažąsias raides.

3.1.3 Eksploatavimo aplinka

Išnagrinėjus panašaus tipo programinę įrangą bei jos reikalavimus sistemoms, planuojami reikalavimai programinei bei aparatūrinei įrangai būtų tokie:

Aparatūrinė įranga:

IBM PC.

Programinė įranga:

Programos greitis priklausys nuo turimos techninės įrangos. Programinės įrangos sukūrimui bei testavimui naudojami Intel® architektūros kompiuteriai. Objektiškai orientuotas programavimas atliekamas Visual Basic kalba.

Kuriant produktą bus naudojama operacinė sistema Windows 98 ir naujesnė, o programinė įranga: Microsoft Visual Studio 6.

3.1.4 Apribojimai

Programos skaičiavimo dalyje numatyti apribojimai:

- programa veiks tik Windows 98 ir naujesnėse versijų Windows aplinkose;
- būtina naudotis pele;
- galima įvesti ne daugiau kaip 100 medžiagų;
- realių skaičių skiriamasis ženklas tarp sveikosios ir trupmeninės dalies nustatomas *Regional Options* sisteminiame dialogo lange;
- tarp įvedamų skaičių negalima palikti tarpo;
- jei apribojimas viršijamas, pasirodo pranešimas apie tai.

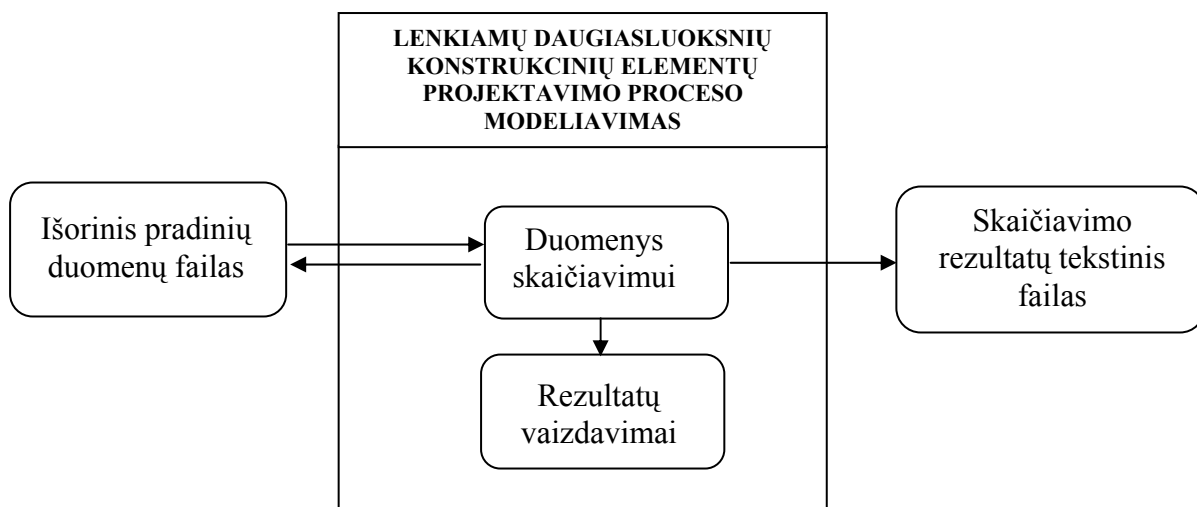
3.1.5 Duomenų srautai

Pradiniai programos duomenys yra konstrukcijos parametrai ir ją veikiančios apkrovos. Skaičiavimo uždavinių realizavimui būtina turėti analitines išraiškas ir mokėti jomis tinkamai pasinaudoti. Todėl būtina teisingai nustatyti:

- neutraliojo sluoksnio koordinatę y_0 ;
- konstrukcinio elemento ekvivalentinį tamprumo modulį E_k ;
- daugiasluoksnių konstrukcijų elementų standumą D ;
- sijos geometrinius parametrus: plotį b ir aukštį h ;
- sluoksnių skaičių n ;
- bet kuriame sijos sluoksnyje:
 - normalinius įtempimus σ_j ;
 - tangentinis įtempimus τ_j ;
 - σ_j įtempimų intensyvumą.

3.2 Duomenų struktūra

Naudoju objektiškai orientuotą programavimo technologiją. Atsižvelgdama į turimus reikalavimus sistemai, pasirenku tokį struktūros modelį (3 pav.).



3 pav. Grafiškai atvaizduotas struktūros modelis

Išorinis pradinių duomenų failas – šiame faile saugomi pradiniai duomenys.

Duomenys skaičiavimui – modulis, skirtas atidaryti anksčiau išsaugotus failus arba įvesti naujus duomenis. Taip pat šiame modulyje atliekamas duomenų koregavimas, medžiagų pasirinkimas, sluoksnių sukeitimas, duomenų tikslinimas bei skaičiavimas pagal formules.

Rezultatų vaizdavimas – modulis, atsakingas už suvestų parametų pateikimą, gautų rezultatų ir grafinių vaizdų pateikimą.

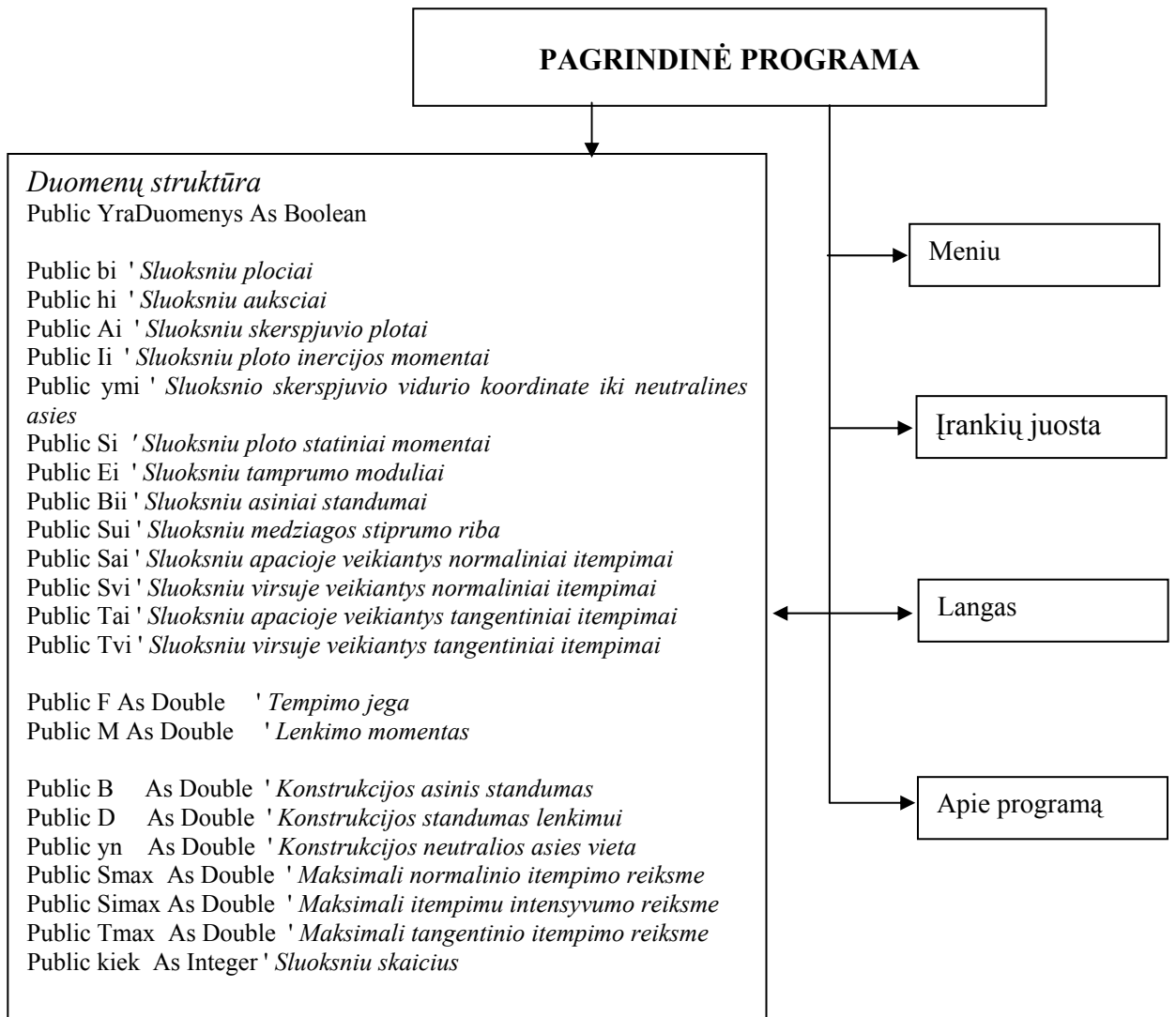
Skaičiavimo rezultatų tekstinis failas – skirtas skaičiavimo duomenų išsaugojimui, jų atspausdinimui.

3.3 Projektuojamos sistemos architektūra

Programos architektūra pavaizduota 4 paveiksle.

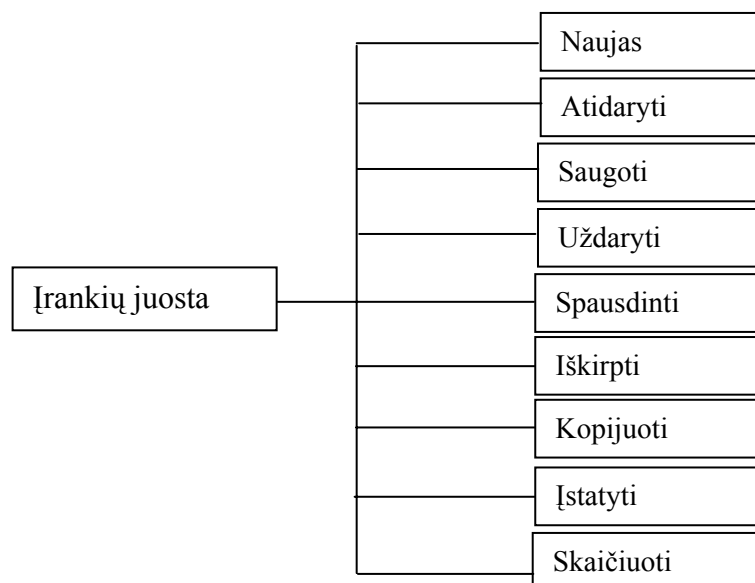
Pagrindiniai sistemos komponentai:

- Duomenų struktūroje aprašomi duomenys;
- Meniu juosta;
- Įrankių juosta;
- Pradinių duomenų ir skaičiavimo rezultatų langas;
- Langas „Apie.. programą”.



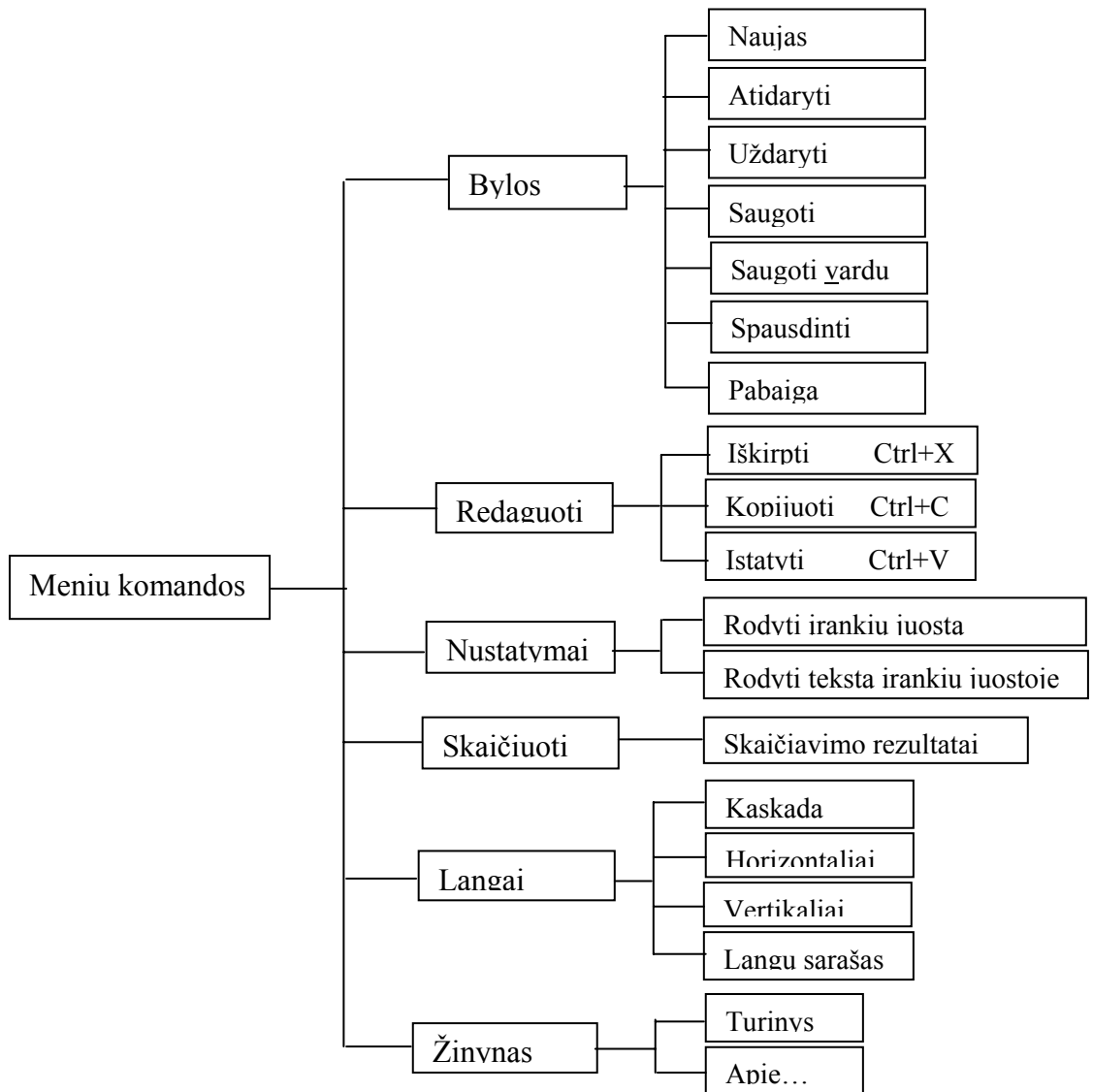
4 pav. Projektuojamos sistemos architektūrinė schema

Įrankių juostos architektūrinė schema pateikta 5 pav. Ją sudaro 9 mygtukai.



5 pav. Komponentės Mygtukai duomenų struktūra

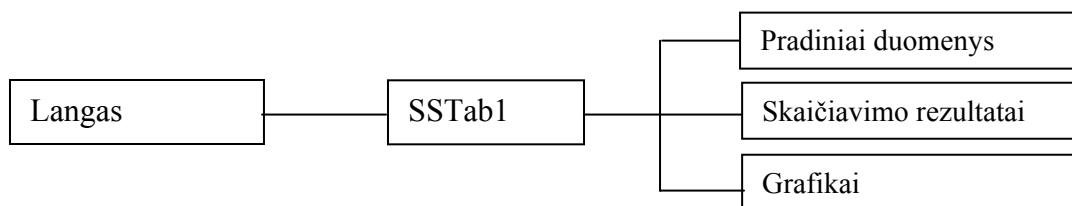
Meniu juostoje yra šeši pagrindiniai vardinimai, kurių kiekvienas dar turi pasirenkamas komandas 6 pav.



6 pav. Komponentės Meniu komandos duomenų struktūra

Lango duomenų struktūra pavaizduota 7 pav. Šiame lange yra trys kortelės:

- pradinių duomenų įvedimas ir redagavimas;
- skaičiavimo rezultatų išvedimas;
- grafinis įtempimų vaizdavimas sluoksnyje



7 pav. Komponentės Langas duomenų struktūra

3.4. Programinių modulių specifikacijos

Sistema buvo išskirta į atskirus modulius.

3.4.1 Modulių pavadinimai

Suprojektuotą sistemą sudaro šie moduliai:

- **Apie...** (Apie.frm);
- **Langas** (Langas.frm);
- **Pagrindas** (Pagrindas.frm);
- **Pradinis** (Vinjete.frm).

3.4.2 Modulių funkciniai aprašymai

Apie...

Šis modulis suteikia informaciją apie programą ir jos autorių.

Aprašymas:

- Paveikslukas – programos logotipas.:
- Parodo programos pavadinimą, autorių, versijos numerį, metus.
- Mygtukai:
 - Informacija apie sistemą – atidaro dialogo langą *System Information*.
 - Gerai – uždaro šį dialogo langą.

Langas

Programos lango Langas aprašymas:

- Trys kortelės:
 - *Pradiniai duomenys:*

- įrašomi apkrovų veikiančių sijos skerspjūvyje dydžiai;
- įtraukiamos į lentelės sąrašą naudojamos medžiagos;
- įrašomi sijos skerspjūvio geometriniai parametrai;
- parodomas grafinis sijos skerspjūvio vaizdas.
- *Skaičiavimo rezultatai.* Parodomi įrašyti:
 - pradiniai duomenys;
 - gauti pagal formules paskaičiuoti skaičiavimo rezultatai.
- *Grafika.* Iš sąrašo pasirenkame, kurių įtempimų epiūras norite matyti:
 - normalinius įtempimus Sigma, MPa;
 - braižyti skerspjūvį;
 - rodyti maksimalias įtempimų reikšmes sluoksniuose;
 - tangentinius įtempimus Tau, MPa;
 - braižyti skerspjūvį;
 - rodyti maksimalias įtempimų reikšmes sluoksniuose;
 - įtempimų intensyvumas Sigma_i, MPa.
 - braižyti skerspjūvį;
 - rodyti maksimalias įtempimų reikšmes sluoksniuose;
 - rodyti medžiagų stiprumo ribas.

– **Pagrindas** (Pagrindas.frm);

Tai pagrindinis programos langas.

Programos lango aprašymas:

– Programos sisteminis meniu (paveiksliukas), programos pavadinimas „Daugiasluoksnių sijų lenkimas“, programos valdymo mygtukai.

– Meniu juosta, kur iš išskleidžiamo sąrašo pasirenkamos komandos. Ši juosta yra darbalaukyje pastoviai.

– Mygtukai – priemonių juosta, kur nuspaudus pasirinktą mygtuką, bus vykdoma komanda.

– Paveiksliukų sąrašas – naudojamas įrankių juostos mygtukų turiniui vaizduoti;

– Sisteminiai dialogo langai.

– **Pradinis** (Vinjete.frm).

Programos logotipas kurį rodo, kol programa užkraunama į atmintį.

3.4.3 Modulio duomenų struktūra

Lango *Pagrindas* duomenų struktūra.

Dinaminiai masyvai:

Public YraDuomenys As Boolean

Public bi ' *Sluoksnių pločiai*

Public hi ' *Sluoksniu aukščiai*

Public Ai ' *Sluoksniu skerspjūvio plotai*

Public Ii ' *Sluoksniu ploto inercijos momentai*

Public ymi ' *Sluoksniu skerspjūvio vidurio koordinate iki netralines ašies*

Public Si ' *Sluoksniu ploto statiniai momentai*

Public Ei ' *Sluoksniu tamprumo moduliai*

Public Bii ' *Sluoksniu asiniai standumai*

Public Sui ' *Sluoksniu medžiagos stiprumo riba*

Public Sai ' *Sluoksniu apačioje veikiantys normaliniai įtempimai*

Public Svi ' *Sluoksniu viršuje veikiantys normaliniai įtempimai*

Public Tai ' *Sluoksniu apačioje veikiantys tangentiniai įtempimai*

Public Tvi ' *Sluoksniu viršuje veikiantys tangentiniai įtempimai*

Kintamieji:

Public F As Double ' *Tempimo jėga*

Public M As Double ' *Lenkimo momentas*

Public B As Double ' *Konstrukcijos asinis standumas*

Public D As Double ' *Konstrukcijos standumas lenkimui*

Public yn As Double ' *Konstrukcijos neutralios ašies vieta*

Public Smax As Double ' *Maksimali normalinio įtempimo reikšme*

Public Simax As Double ' *Maksimali įtempimu intensyvumo reikšme*

Public Tmax As Double ' *Maksimali tangentinio įtempimo reikšme*

Public kiek As Integer ' *Sluoksniu skaičius*

Lango *Langas* struktūra

Naudojama vartotojo sąsajos prasukimo juostų išpildymui.

Public Top1, Top2, Top3, Left1 As Integer.

3.4.4 Modulio sąsaja, pasirinkta dialogo struktūra

Programos „Daugiasluoksnių sijų lenkimas” sistemos modelio langų *Pradiniai duomenys*, *Skaičiavimo rezultatai*, *Grafikai* dialogo struktūra.

Lango **Pradiniai duomenys** struktūra

Šis langas suskirstytas į tris skyrius:

- APKROVOS SKERSPJŪVYJE;
- SIJOS MEDŽIAGOS;
- SIJOS GEOMETRIJA.

Skyrius APKROVOS SKERSPJŪVYJE

Aprašymas:

- Lenkimo momento įrašą M – reikšmės įrašomos tekstiniame laukelyje;
- Skersinės jėgos įrašą F - reikšmės įrašomos tekstiniame laukelyje.

Skyrius SIJOS MEDŽIAGOS

Aprašymas:

- Medžiagos pavadinimas – medžiagos pavadinimai numeruojamos sužymint sveikais skaičiais. Reikšmės įrašomos tekstiniame laukelyje.
- Medžiagos tamprumo modulis E – taip pat rodomas medžiagos numeris, o reikšmės įrašomos tekstiniame laukelyje.
- Medžiagos stiprumo riba S_u – rodomas medžiagos numeris, tekstas įrašomas tekstiniame laukelyje. Neįrašius šio dydžio skaičiavimai bus vykdomi.
- Mygtukai:
 - Pridėti medžiagą.
 - Pašalinti medžiagą.
- Lentelė, kurioje suvestos medžiagos.

Skyrius SIJOS GEOMETRIJA

Aprašymas:

- Įvesta medžiaga pasirenkama iš sąrašo arba lentelės.

- Pasirinktos medžiagos sluoksnio parametrai:
 - plotis b ;
 - aukštis h .
- Mygtukai:
 - Pridėti sluoksnį.
 - Pašalinti sluoksnį.
- Lentelė, kurioje suvestos medžiagos.
- Sluoksnių sukeitimas vietomis.
- Grafinis sijos skerspjūvio vaizdas.

Skyrius SKAIČIAVIMO REZULTATAI

Parodomi suvesti duomenys ir gauti skaičiavimų rezultatai.

Skyrius GRAFIKAI

Pagal atliktus skaičiavimų rezultatus nubraižomos įtempimų epiūros:

- normalinius įtempimus Σ , MPa;
- tangentinius įtempimus τ , MPa;
- įtempimų intensyvumas Σ_i , MPa

3.5 Testavimas

Testavimo metu keliami tokie uždaviniai:

- patikrinti programos išeities teksto teisingumą;
- patikrinti, ar programa teisingai atlieka reikalingus skaičiavimus ir pateikia jų rezultatus.

Sprendžiant pirmąjį uždavinį atliekamas išeities kodo teisingumo tikrinimas. Tai atlieka *Visual Basic* integruota kūrimo aplinka, programos kodą sukompiluoja. Suradus klaidingas programos išeities kodo vietas, kompiliatorius išvesdavo klaidingos kodo vietos eilutės numerį, kurioje įvyko klaida, klaidos aprašymą. Toks išeities kodo testavimas tęsiasi tol, kol kompiliatorius neberanda nė vienos sintaksės klaidos, kas leistų programai veikti be sutrikimų.

Antrasis testavimo etapas buvo svarbus daugiasluoksnių kompozicinių konstrukcinių elementų sistemos modelio sėkmingo veikimo užtikrinimui. Kiekviena funkcija ar procedūra savyje turi kodą. Tam, kad įvykus nenumatytai klaidai programa bandytų tęsti darbą, kiekviena funkcija pradedama eilute *On Error Resume Next*.

Įvedus neteisingus (neteisingas formatas) skaitinius duomenis programos išveda pranešimą apie neteisingą formatą. Nuskaitant pradinius duomenis iš failo, kurio neteisingas formatas, išvedamas pranešimas apie neteisingus duomenis faile. Vykdamas spausdinimą spausdinimo klaidos perduodamos operacinei sistemai.

Skaičiavimo teisingumui patikrinti atliekamas testavimas atskirų komponentių.

- Knygoje [4] laboratorinių darbų pavyzdžių atsakymų rezultatus sulyginu su gautais programos paskaičiuotais rezultatais.
- Suvestų rankomis duomenų paskaičiuotų rezultatų palyginimas su programos skaičiavimo rezultatais. Gautus rezultatus palyginame.
- Savo pratestuotą programą duodu testuoti ekspertui. Ekspertas pratestavęs pateikia patvirtinimą, kad programa skaičiuoja teisingai.

Žemiau pateikiamas vieno uždavinio testavimo pavyzdys:

Sąlyga

Sijos skerspjūvio matmenys nurodyti milimetrais. Medienos parametrai: $E_1=1,25 \cdot 10^4$ MPa, $\sigma_{adm.t}=40$ MPa, $\sigma_{adm.c}=20$ MPa, $\tau_{adm.s}=3,0$ MPa; plieno parametrai: $E_2=20 \cdot 10^4$ MPa, $\sigma_{adm.tc}=160$ MPa, $\tau_{adm}=80$ MPa ir sąlyčio zonoje $\tau_{adm.s}=2,5$ MPa. Sijos plotis 120 mm, Pirmos medžiagos aukštis 20 mm, antros – 200 mm. Patikrinti sijos stiprumą.

- Pavyzdyje [2] pateikti paskaičiavimai:

$$B_1 = 1,25 \cdot 10^{10} \cdot 0,12 \cdot 0,2 = 300 \text{ MN},$$

$$B_2 = 2 \cdot 10^{10} \cdot 0,12 \cdot 0,2 = 480 \text{ MN},$$

$$B = 780 \text{ MN},$$

$$\gamma_0 = \frac{(300 \cdot 0,2 + 480 \cdot 0,02 + 2 \cdot 480 \cdot 0,2) \cdot 10^6}{2 \cdot 780 \cdot 10^6} = 0,168 \text{ m},$$

$$\gamma_1 = \gamma_0 - 0,5 \cdot \delta_1 = 0,168 - 0,100 = 0,068 \text{ m},$$

$$\gamma_2 = \gamma_0 - \delta_1 - 0,5 \cdot \delta_2 = 0,168 - 0,2 - 0,01 = -0,042 \text{ m}.$$

$$I_1 = \frac{0,12 \cdot 0,2^3}{12} + 0,12 \cdot 0,2 \cdot 0,068^2 = 1,9 \cdot 10^{-4} \text{ m}^4,$$

$$I_2 = \frac{0,12 \cdot 0,02^3}{12} + 0,12 \cdot 0,02 \cdot 0,042^2 = 0,043 \cdot 10^{-4} \text{ m}^4,$$

$$D = 1,25 \cdot 10^{10} \cdot 1,9 \cdot 10^{-4} + 20 \cdot 10^{10} \cdot 0,043 \cdot 10^{-4} = 3,24 \text{ MNm}^2.$$

$$\sigma_{2v} = 40 \cdot 10^3 \cdot 0,168 \cdot 1,25 \cdot 10^{10} / 3,24 \cdot 10^6 = 25,9 \text{ MPa},$$

$$\sigma_{2a} = 40 \cdot 10^3 \cdot (-0,032) \cdot 1,25 \cdot 10^{10} / 3,24 \cdot 10^6 = -4,9 \text{ MPa},$$

$$\sigma_{1v} = \sigma_{2a} \cdot E_2 / E_1 = -4,9 \cdot 20 / 1,25 = -79,0 \text{ MPa},$$

$$\sigma_{1a} = 40 \cdot 10^3 \cdot (-0,052) \cdot 20 \cdot 10^{10} / 3,24 \cdot 10^6 = -128,4 \text{ MPa}.$$

$$C_I=0,$$

$$C_I=1,25 \cdot 10^{10} \cdot (0,168 \cdot 0,12) \cdot 0,168 / 2 = 21 \text{ MNm},$$

$$C_5=0,$$

$$C_4=C_3=20 \cdot 10^{10} (0,12 \cdot 0,02) \cdot 0,042 = 20 \text{ MNm},$$

$$\tau_{1a} = \tau_{1v} = 0,$$

$$\tau_{1v} = 50 \cdot 10^3 \cdot 21 \cdot 10^6 / 3,24 \cdot 10^6 \cdot 0,12 = 2,70 \text{ MPa},$$

$$\tau_{2a} = 50 \cdot 10^3 \cdot 20 \cdot 10^6 / 3,24 \cdot 10^6 \cdot 0,12 = 2,56 \text{ MPa}.$$

- Skaičiavimai atlikti naudojantis programa. Programa pagal duomenis pateiktus uždavinio sąlygoje išveda sekančius rezultatus:

PRADINIAI DUOMENYS

Lenkimo momentas $M = 40000000,00 \text{ [N mm]}$

Skersinė jėga $F = 50000,00 \text{ [N]}$

1-o sluoksnio plotis $b[1] = 120,00 \text{ [mm]}$

1-o sluoksnio aukštis $h[1] = 20,00 \text{ [mm]}$

1-o sluoksnio medžiagos tamprumo modulis $E[1] = 200000,00 \text{ [MPa]}$

2-o sluoksnio plotis $b[2] = 120,00 \text{ [mm]}$

2-o sluoksnio aukštis $h[2] = 200,00 \text{ [mm]}$

2-o sluoksnio medžiagos tamprumo modulis $E[2] = 12500,00 \text{ [MPa]}$

Pastaba: cia pirmas sluoksnis yra apatinis, paskutinis - viršutinis.

SKAICIAVIMO REZULTATAI

Konstrukcijos ašinis standumas $B = 780000000,000$ [N]

Konstrukcijos standumas lenkimui $D = 3249846153846,150$ [N mm²]

Neutrales padetis nuo skerspjūvio apacios $y_n = 52,308$ [mm]

Normaliniai itempimai S [MPa]:

$$S_a[1] = -128,763$$

$$S_v[1] = -79,530$$

$$S_a[2] = -4,971$$

$$S_v[2] = 25,800$$

Tangentiniai itempimai T [MPa]:

$$T_a[1] = 0,000$$

$$T_v[1] = 2,604$$

$$T_a[2] = 2,604$$

$$T_v[2] = 0,000$$

Itempimu intensyvumas S_i [MPa]:

$$S_{i_a}[1] = 128,763$$

$$S_{i_v}[1] = 79,658$$

$$S_{i_a}[2] = 6,712$$

$$S_{i_v}[2] = 25,800$$

PASTABA: cai indeksas 'a' reiškia apatine sluoksniu plokštuma, o indeksas 'v' - viršutine.

Maksimalios itempimu reikšmės

$S_{max} = -128,763$ [MPa] - 1 sluoksnyje.

$T_{max} = 2,704$ [MPa] - 2 sluoksnyje.

$S_{imax} = 128,763$ [MPa] - 1 sluoksnyje.

Galima pastebėti, kad skaičiavimo rezultatai gauti pavyzdyje ir suskaičiavus programa, skiriasi nuo 0,15% iki 1,7%. Tai galima paaiškinti tuo, kad programa skaičiuoja naudodama dvigubo tikslumo realius skaičius (penkiolikos skaičių tikslumu), o pavyzdyje skaičiuota tik šimtųjų tikslumu.

4. VARTOTOJO DOKUMENTACIJA

4.1. Funkcinis sistemos aprašymas

Programos paskirtis - leisti vartotojui su bet kuriuo kompiuteriu, kuriame yra Windows 98 arba naujesnė versija atlikti lenkiamų daugiasluoksnių konstrukcinių elementų skaičiavimus.

Pagrindinės sistemos savybės:

- Paprastas valdymas.
- Parametrai skaičiuojami dvigubu tikslumu.
- Grafiškas konstrukcijos atvaizdavimas.
- Duomenų išsaugojimas, spausdinimas.










4.2. Sistemos vadovas

– Meniu

- **Bylos**. Galima pasirinkti sekančias komandas:
 - **Naujas** (*Ctrl+N*) - kurti naują darbą.
 - **Atidaryti** (*Ctrl+O*) – įkelti anksčiau sukurtą darbą.
 - **Uždaryti** - uždaro naujo darbo langą.
 - **Saugoti** – išsaugoti sukurtą darbą byloje.
 - **Saugoti_vardu** - išsaugoti sukurtą darbą byloje kitu vardu.
 - **Spausdinti** – spausdinti skaičiavimo rezultatus ir grafikus.
 - **Pabaiga** – išeiti iš sistemos.
- **Redaguoti**. Pasirenkamos komandos:
 - **Iškirpti** (*Ctrl+X*) – iškirpti duomenis.
 - **Kopijuoti** (*Ctrl+C*) – kopijuoti duomenis į laikinąją atmintį.
 - **Išstatyti** (*Ctrl+V*) – įkelti pradinius duomenis iš laikinosios atminties.
- **Nustatymai** – pasirenkami įrankių juostos nustatymai:
 - **Rodyti įrankių juostą** – pasižymėjus - įrankių juosta bus matoma programos lange, o atšaukus - ji bus paslėpta.
 - **Rodyti tekstą įrankių juostoje** – po paveiksliukais bus matomas įrankio pavadinimas, atšaukus komandą – bus tik paveiksliukai.
- **Skaičiuoti** – atliekami skaičiavimo veiksmai.
- **Langai** – darbas su keletu atidarytų langų:

- **Kaskada** – išdėstyti atidarytus langus kaskadomis.
- **Horizontaliai** - išdėstyti atidarytus langus horizontaliai.
- **Vertikaliai** - išdėstyti atidarytus langus vertikaliai.
- **Sąrašas atidarytų darbo langų.**
- **Žinynas** - vartotojo pagalba.
 - **Turinys** – paleidžia pagalbos sistemą.
 - **Apie...** - – iššaukia dialogą langą „Apie...”

– Įrankių juosta

	Naujas darbas		Panaikinti duomenis
	Įkelti anksčiau sukurtą darbą		Nukopijuoti į laikinąją atmintį
	Išsaugoti darbą		Įkelti iš laikinosios atminties
	Uždaryti naują langą		Skaičiuoti
	Spausdinti skaičiavimo rezultatus		

– Pradiniai duomenys

Medžiagos įvedimas, redagavimas ir trynimasis

- Vartotojas į tekstinius laukelius įrašo:
 - medžiagų pavadinimus;
 - apkrovos dydžius:
 - lenkimo momento įrašos M reikšmę;
 - skersinė jėgos įrašos F reikšmę;
 - sijos geometrinius parametrus.
- Teisingai užpildomi žemiau išvardinti laukai:
 - tamprumo modulis E ;
 - medžiagos stiprumo ribą S_u ;

ir spaudžiamas mygtukas „Pridėti medžiagą”.

- Redagavimui pasižymima medžiaga, atliekami pakeitimai ir spaudžiame *Enter*.
- Redagavimui - iškirpti, kopijuoti, įklijuoti - galima naudoti standartines Windows aplinkos komandas *Cut, Copy, Paste*.
- Trynimui – pasižymima medžiaga ir spaudžiamas mygtukas „Pašalinti medžiagą” arba klavišas *Delete*.

Pradiniai duomenys | Skaičiavimo rezultatai | Grafikai

APKROVOS SKERSPJŪVIJJE

Lenkimo momento įrašas $M = 100.15$ [Nm]
 Skersinė jėgos įrašas $F = 10000$ [N]

SIJOS MEDIJAIAGOS

3 medžiagos pavadinimas: Šalto kietėjimo stiklo plastikas
 3 medžiagos tamprumo modulis $E[3] = 160$ [MPa]
 3 medžiagos stiprumo riba $Su[3] = 240$ [MPa] **nebūtinus**

Nr.	Medžiagos pavadinimas	E, MPa	Su, MPa
1	Anglies plastikas	260	360.7
2	Karšto kietėjimo stiklo plastikas	200.22	150.5
3	Šalto kietėjimo stiklo plastikas	160	240

8 pav. Lango Pradiniai duomenys grafinis vaizdas

Sluoksnio įvedimas, redagavimas ir trynimasis

- Teisingai įrašomi sijos skerspjūvio geometriniai parametrai:
 - sluoksnio plotis b ;
 - sluoksnio aukštis h ;
 - pasinaudojus prasukimo juosta iš surašyto sąrašo pasirenkama sluoksnio medžiaga;
- Sluoksniui pridėti spaudžiamas mygtukas „Pridėti sluoksnį“.
- Sluoksniui pašalina spaudžiamas mygtukas „Pašalinti sluoksnį“. Prieš tai nereikalingą sluoksnį reikia pasižymėti.
- Trynimui – pasižymėjimas medžiaga ir spaudžiamas mygtukas „Pašalinti medžiagą“ arba klavišas *Delete*.
- Redagavimui sluoksnio parametų (iškirpti, kopijuoti, įklijuoti) galima naudoti standartines Windows aplinkos komandas *Cut*, *Copy*, *Paste*.
- Sluoksnis galima sukeisti vietomis.
- Pateikiamas sijos skerspjūvio vaizdas.
- Sukeitus sluoksnius vietomis, grafinis skerspjūvio vaizdas taip pat pasikeis.

- Failas išsaugomas formatu *.lsd (lenkiamų sijų duomenys).

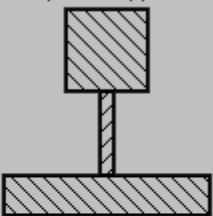
SIJOS SKERSPJŪVIO GEOMETRIJA

2 sluoksnio plotis b[2] = [mm]

2 sluoksnio aukštis h[2] = [mm]

2 sluoksnio medžiaga:

Sijos skerspjūvis



Sluoksniai numeruojami iš apačios į viršų.

Nr.	b, mm	h, mm	Medžiaga
1	15	3	Šalto kietėjimo stiklo plastikas
2	1	6	Šalto kietėjimo stiklo plastikas
3	6	6	Karšto, kietėjimo stiklo plastikas

9 pav. Lango Pradiniai duomenys sijos skerspjūvio grafinis vaizdas

– Skaičiavimo rezultatai

Pasirinkus meniu komandą arba nuspaudus įrankį **Skaičiuoti**, pagal formules bus atliekami skaičiavimai. Rezultatai parodomi kortelėje *Skaičiavimo rezultatai*:

- pradiniai duomenys;
- skaičiavimo rezultatai.
- galima išsaugoti:
 - kaip tekstinį failą formatu *.txt;
 - taip pat nukopijuoti duomenis į tekstinį redaktorių;
 - atspausdinti.

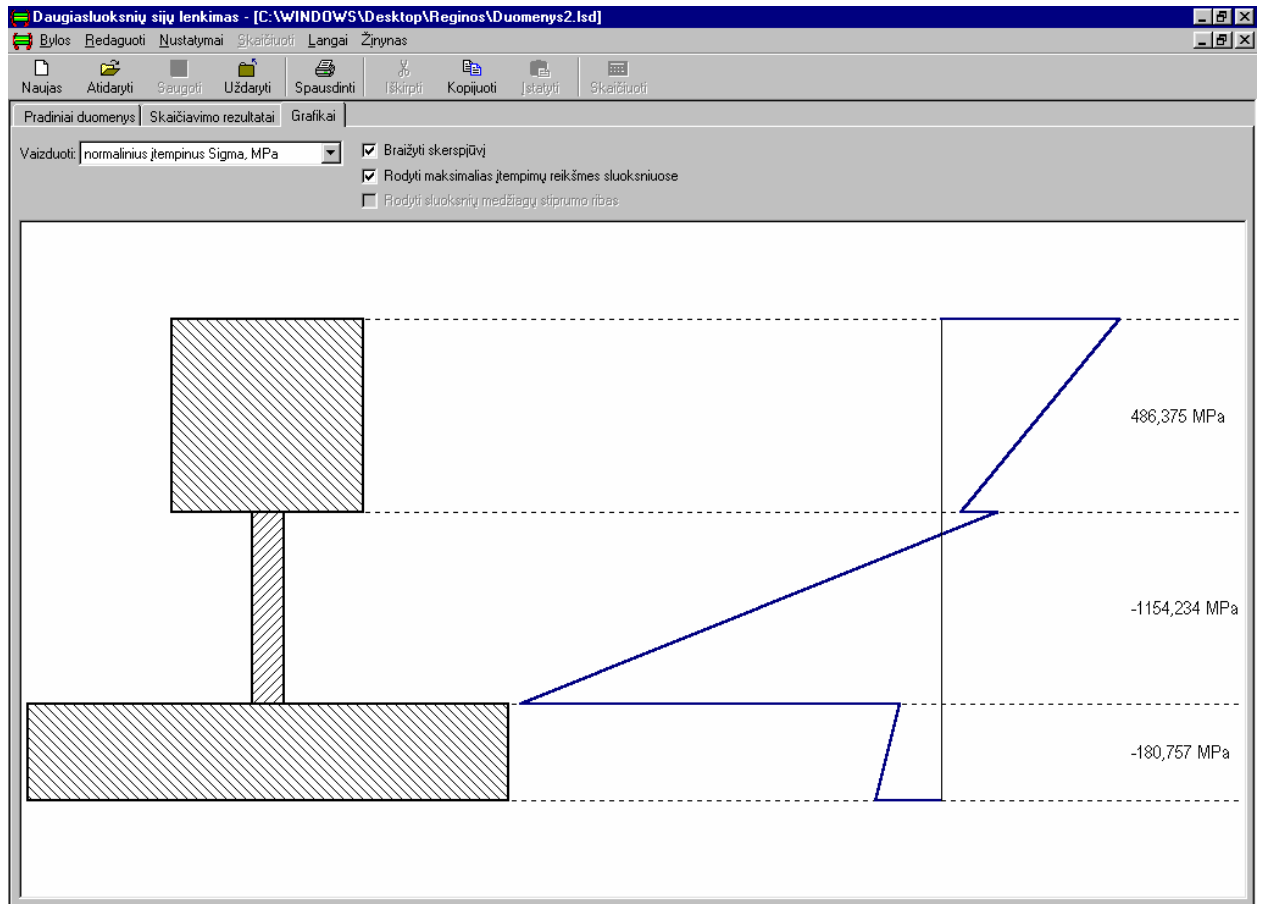
– Grafikai

- Iš sąrašo pasirenkame, kurių įtempimų epiūras norite matyti:
 - normalinius įtempimus Sigma, MPa;
 - tangentinius įtempimus Tau, MPa;
 - įtempimų intensyvumas Sigma_i, MPa.

Galimi pasirinkimų nusistatymai:

- braižyti skerspjūvį;

- rodyti maksimalias įtempimų reikšmes absoliutinių dydžių sluoksniuose;
- rodyti sluoksnių medžiagų stiprumo ribas.



10 pav. Lango Grafikai skaičiavimo rezultatų grafinis atvaizdavimas

Programos lange *Pradiniai duomenys* srityse: *Apkrovos*, *Sijos medžiagos*, *Sijos geometrija* tekstiniuose laukeliuose programa leidžia įvesti, keisti ir trinti duomenis. Jei įvesti duomenys klaidingai, programa atskiru dialogo langu informuoja apie padarytas klaidas šiais pranešimais:

- tamprumo modulis negali būti mažesnis už 0! Prašome pataisyti.
- sluoksnio plotis b negali būti mažesnis arba lygus 0! Prašome pataisyti.
- sluoksnio aukštis h negali būti mažesnis arba lygus 0! Prašome pataisyti.
- prašome parinkti sluoksnio medžiagą.
- medžiaga tokiu pavadinimu jau egzistuoja. Prašome pataisyti.
- stiprumo riba modulis negali būti mažesnis už 0. Prašome pataisyti.

Teisingai įrašius duomenis, atliekami skaičiavimai, bei parodomas grafinis sijos skerspjūvio vaizdas ir jame veikiančys įtempimai.

Vartotojas gali greitai pamatyti skaičiavimų rezultatus pasirinkęs meniu *Skaičiuoti*, spausdamas mygtuką *Skaičiuoti*, bei pasirinkęs kortelę *Skaičiavimo rezultatai*. Jei įvesti ne visi reikalingi duomenys pasirodo pranešimų dialogo langai:

- prieš skaičiavimą reikia užduoti apkrovą.
- neteisingai įvesta skersinė jėga! Prašome pataisyti.
- neteisingai įvestas lenkimo momentas! Prašome pataisyti.
- skaičiavimo metodika reikalauja bent dviejų sluoksnių.

Neatlikus reikalavimų pranešimuose, negalėsite gauti skaičiavimo rezultatų.

Skaičiavimų rezultatai išvedami atskirame lange. Čia galima peržiūrėti įvestus pradinius duomenų parametrus. Kiekvieno sluoksnio atskirai išvedami rezultatai: įtempimų skaitinės reikšmės sluoksnio viršuje ir apačioje, *Max* įtempimų dydis bei sluoksnis ir kuriame jie kyla lenkimo neutralės koordinatė ir standumas lenkimui. Pasirinkus meniu *Redaguoti* arba nuspaudus mygtuką *Kopijuoti*, pasižymėtus duomenis galima nukopijuoti ir perkelti į tekstinio redaktoriaus programą.

Atlikus skaičiavimus, jei pasirinktos medžiagos stiprumo riba didesnė už leistina riba, prie skaičiavimo rezultatų matysis pranešimas „VIRŠYTA STIPRUMO RIBA!!!”

SKAIČIAVIMO REZULTATAI

Konstrukcijos ašinis standumas $B = 1026000,000$ [N]
Konstrukcijos standumas lenkimui $D = 8545500,000$ [N mm²]
Neutralės padėtis nuo skerspjūvio apačios $y_n = 10,500$ [mm]

Normaliniai įtempimai S [MPa]:

$Sa[1] = -740,916$
 $Sv[1] = -317,536$

$Sa[2] = -317,536$
 $Sv[2] = -105,845$

$Sa[3] = -352,817$
 $Sv[3] = 1058,452$

Tangentiniai įtempimai T [MPa]:

$Ta[1] = 0,000$
 $Tv[1] = 315,972$

$Ta[2] = 21,065$
 $Tv[2] = 84,259$

$$Ta[3] = 210,648$$
$$Tv[3] = 0.000$$

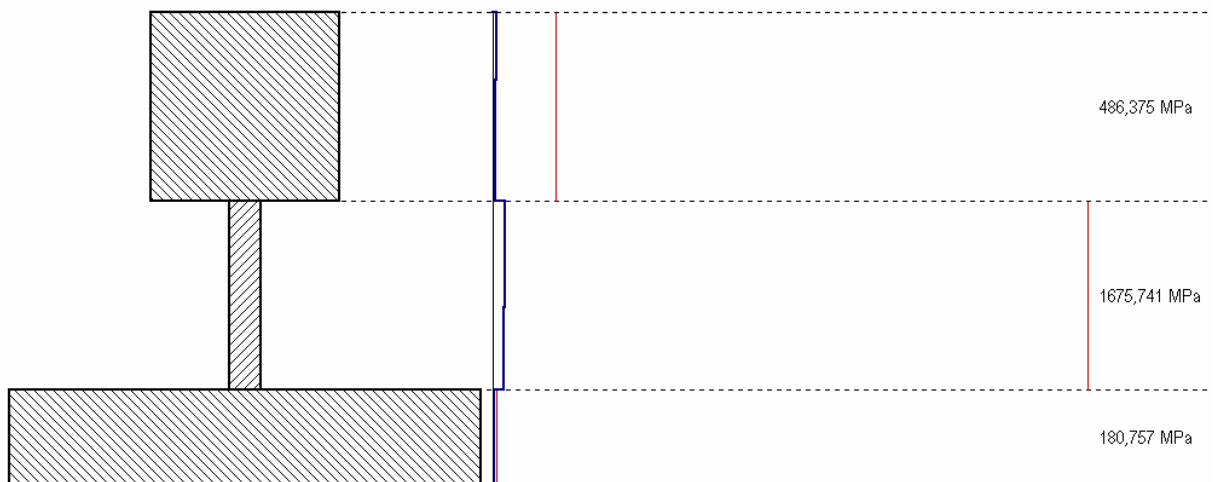
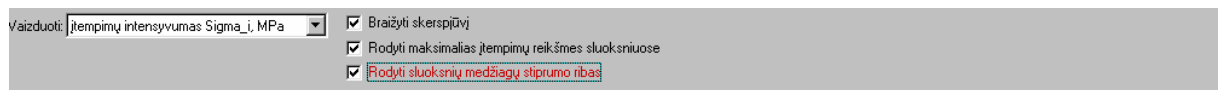
Įtempimų intensyvumas σ_i [MPa]:

$$\sigma_{i_a}[1] = 740,916 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$
$$\sigma_{i_v}[1] = 632,727 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$

$$\sigma_{i_a}[2] = 319,625 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$
$$\sigma_{i_v}[2] = 180,283 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$

$$\sigma_{i_a}[3] = 507,541 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$
$$\sigma_{i_v}[3] = 1058,452 \text{ VIRŠYTA STIPRUMO RIBA !!!}$$

Pasirinkus kortelę *Grafikai* iš sąrašo galima pasirinkti įtempimų grafikus. Grafiškai pavaizduoja kaip pasiskirsto tangentiniai ir normaliniai įtempimai sluoksniuose bei įtempimų intensyvumas. Išvedama absoliučiai didžiausia sluoksnyje veikianti įtempimų reikšmė.



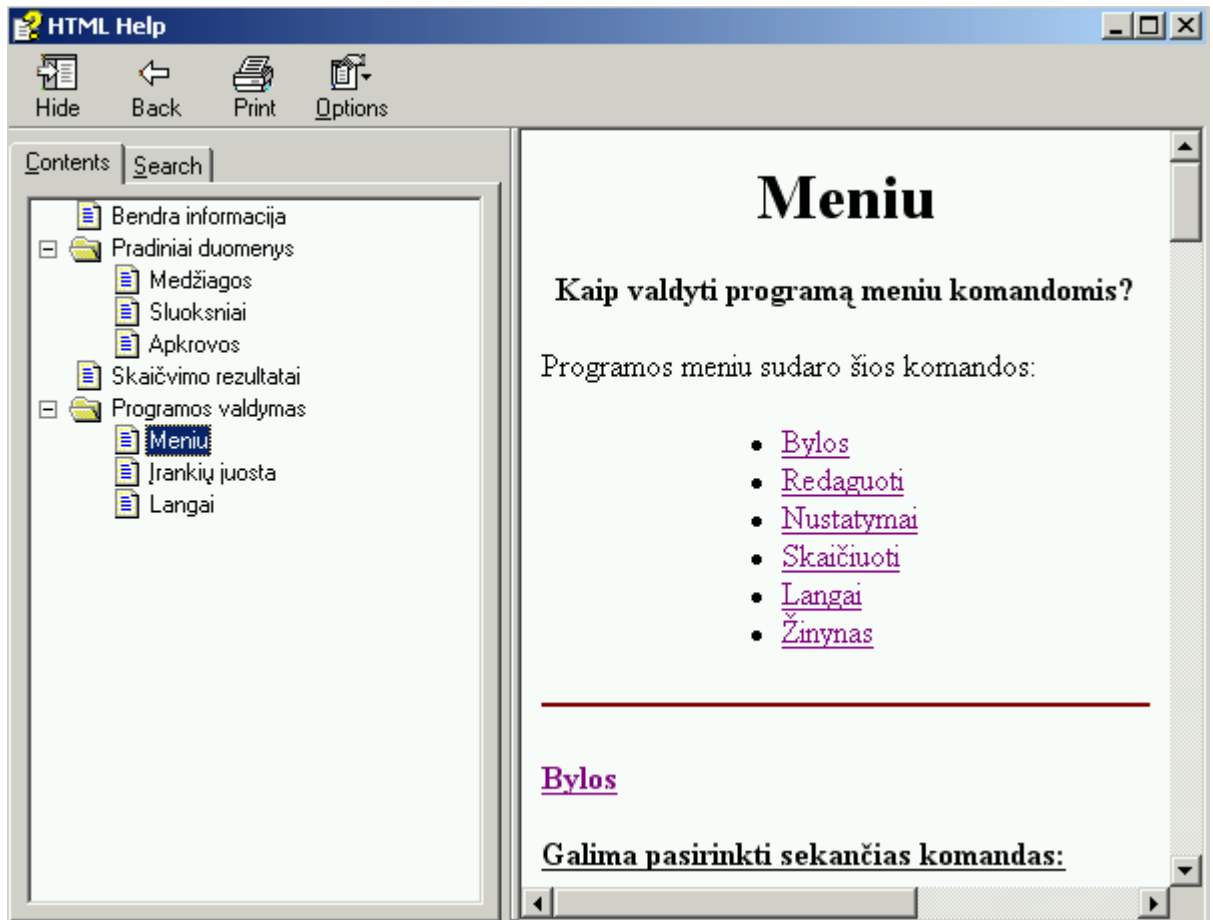
11 pav. Grafiškai pavaizduoti didžiausi sluoksnyje veikiantys įtempimai

Nuėmus pirmą varnelę *Braižyti skerspjūvi*, nebus rodomas skerspjūvis. Nuėmus antrą varnelę *Rodyti maksimalias įtempimų reikšmes sluoksniuose*, nebus rodomos maksimalios reikšmės.

Su meniu *Bylos* pasirinkimais galima atlikti tokius pačius veiksmus kaip ir paspaudus atitinkamą mygtuką: *Saugoti*, *Uždaryti*, *Atidaryti*, *Naujas*. Atliktų skaičiavimų duomenis galima išsaugoti tekstiniame faile nurodytame aplankale.

Vartotojas skaičiavimų duomenis, įtempimų epiūrą gali atsispausdinti pasirinkęs meniu *Byla*⇒*Spausdinti* arba spausdamas mygtuką *Spausdinti*.

Pasirinkus meniu *Žinynas* galima išsikviesti pagalbą, ir iš sąrašo pasirinkti *Turinys*.



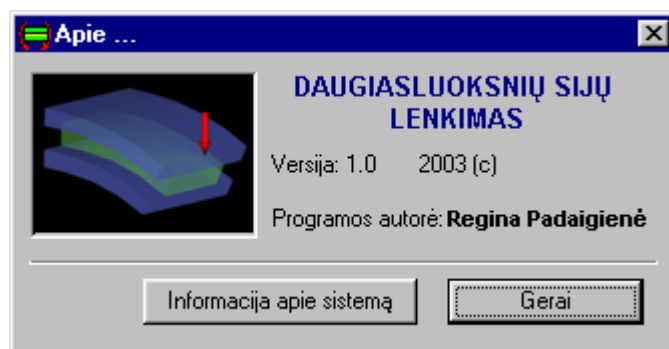
12 pav. Dialogo lango *Turinys* grafinis vaizdas

Čia galima rasti reikiamus paaiškinimus pasirinkus:

- *Bendra informacija* – rasite bendrą informaciją apie programą (kam skirta, kas autorius).
 - Pasirinkę nuorodą teorija, rasite skaičiavimo metodikos aprašymą (pagal kokias formules skaičiuoja, kokie apribojimai ir pan.);
- *Pradiniai duomenys* – galima pasiskaityti apie:
 - *Medžiagas* (jų įvedimą, redagavimą, trynimą);

- *Sluoksniai* (įvedimas, redagavimas, trynimas);
 - *Apkrovos* (įvedimas, redagavimas).
- *Skaičiavimo rezultatai* – pateikiami, kokie skaičiavimo rezultatai gali būti gauti, ką su jais galima daryti ir kaip (saugoti, kopijuoti, spausdinti).
- Programos valdymas:
- *Langai* – rasite informaciją apie programos langus, kam ir kokia tų langų paskirtis;
 - *Meniu* – pateikiami aprašymai apie meniu langus;
 - *Įrankių juosta* – įrankių juostos ir mygtukų aprašymai.

Pasirinkus meniu *Apie...*, atsidaro dialogo langas „*Apie...*“, kuriame yra du mygtukai. Nuspaudus mygtuką „*Informacija apie sistemą*“ gausite informacijos apie sistemą, o nuspaudę „*Gerai*“ uždaro šį dialogo langą.



13 pav. Dialogo langas *Apie...*

4.3 Sistemos instaliavimo dokumentas

Sistemos instaliavimo dokumentas apima: programinės įrangos reikalavimus, sistemą sudarančius failus, minimali reikalingos techninės įrangos konfigūracija.

4.3.1. Programinės įrangos reikalavimai:

- Operacinė sistema: Windows 95/98/ME/2000/XP;
- Papildoma programa Window Installer Package.

4.3.2. Sistemą sudarantys failai

Jie bus tame kataloge į kurį bus instaliuota programa.

- DSL.EXE;
- MSHFLXGD.OCX;
- RICHTX32.OCX;
- DSL.CHM.

4.3.3 Techninės įrangos minimalūs reikalavimai:

- Papildomos vietos kietame diske apie 2 MB
- Tokie pat reikalavimai, kaip Windows 98.

4.3.4 Modelio instaliavimas

Pasirenkamas paketo instaliavimo failas *Install_dsl.exe*. Instaliuojant nuspaudus mygtuką „Toliau“ programa susiinstaliuoja automatiškai. Instaliacijos metu galima nurodyti katalogą į kurį bus instaliuojama programa. Po instaliacijos programos paleidimas *Start⇒Programs⇒Medžiagų mechanika⇒Daugiasluoksnių sijų lenkimas*.

PRODUKTO KOKYBĖS ĮVERTINIMAS

PĮ sistemos kokybė bus vertinama sistemos testavimo metu, bei įvertinant vartotojų atsiliepimus (anketos vartotojams).

PAGRINDINIAI REZULTATAI IR IŠVADOS

Daugiasluoksnių konstrukcinių elementų panaudojimas yra efektyvus būdas pagerinti gaminio technines charakteristikas. Išnagrinėjus lenkiamų daugiasluoksnių konstrukcijų elementų skaičiavimo metodikų analizę, galima padaryti tokias išvadas:

1. Daugiasluoksnių konstrukcinių elementų projektavimo proceso modeliavimas yra paprastas ir nesudėtingas uždavinys, kuris gali būti kompiuterizuotas, sukuriant modeliavimo sistemą. Šioje sistemoje naudojama visiems įprasta grafinė vartotojo sąsaja.
2. Atliekami tikslūs lenkiamų daugiasluoksnių konstrukcijų elementų, kurie yra veikiami skersinių jėgų ir lenkimo momentų, standumo ir stiprumo skaičiavimui.
3. Keičiant įvairius parametrus galima modeliuoti įvairias sijų konstrukcijas. Dėl savo paprastumo lengvai suprantamas kiekvienam mechanikos srities studentui, inžinieriui. Šis projektavimo modelis gali būti taikomas moksliniams tyrimams, inžineriniams skaičiavimams, laboratorinių bei kursinių projektų skaičiavimams.
4. Programos pagalba galima įvertinti lenkiamo daugiasluoksnio konstrukcinio elemento stiprumą: jei bent viename sluoksnyje įtempimų intensyvumas viršija to sluoksnio medžiagos stiprumo ribos (leistinuosius) įtempimus, tai konstrukcija neatlaikys nurodytos išorinės apkrovos. Tokiu atveju programa išveda pranešimą.
5. Pagal pradinius duomenis apskaičiuotų įtempimų reikšmės grafiškai pavaizduojamos ir parodo jų pasiskirstymą daugiasluoksnių konstrukcinių elementų skerspjūvyje, tuo ženkliai palengvinant projektuotojo darbą.

LITERATŪRA:

1. Bareišis J., Paulauskas V. Daugiasluoksnių kompozicinių konstrukcinių elementų projektavimas. – vadovėlis K.: Technologija, 1995.
2. Čižas A.. Medžiagų atsparumas (Konstrukcinių elementų mechanika). V.Technika. 1993.
3. Ostreika A. Programavimo Visual Basic pagrindai: mokojoji knyga. – K.: Technologija, 2003. –225 p.
4. Bareišis J.,Tautkus A., Roličius J. Medžiagų mechanikos ir konstrukcinių elementų atsparumo laboratoriniai darbai. – K.: Technologija, 2000. – 48 p.
5. Garuckas D. Sluoksniuotų kompozicinių konstrukcinių elementų stiprumo ir standumo tyrimai. Daktaro disertacija, 2000, 117 p.
6. Bareišis J., Garuckas D. Daugiasluoksnių strypų standumas ir stiprumas. *Mechanika*, 2000, Nr. 4 (24), - 23-29 p.
7. V.Keršienė, R.Misevičienė, R.Sturienė. Microsoft FrontPage 2000, K.:Technologija, 2002. – 96 p.
8. V.Keršienė, R.Misevičienė, K.Simonavičius,R.Sturienė. Tinklo svetainių projektavimas MS FrontPage aplinkoje, K.:Technologija, 2003. – 98 p.
9. Справочник по композиционным материалам. Ред. Д.Любин, т. 2, М., 1988, 579 с.
10. Banc L.C. Shear Coefficients for Thin Welded Composite Beams. *Composite Structures*, v. 8, 1987, p. 47-61.
11. Васильев В. Механика конструкции из композитных материалов. Москва, 1988, 772 с.

SUMMARY

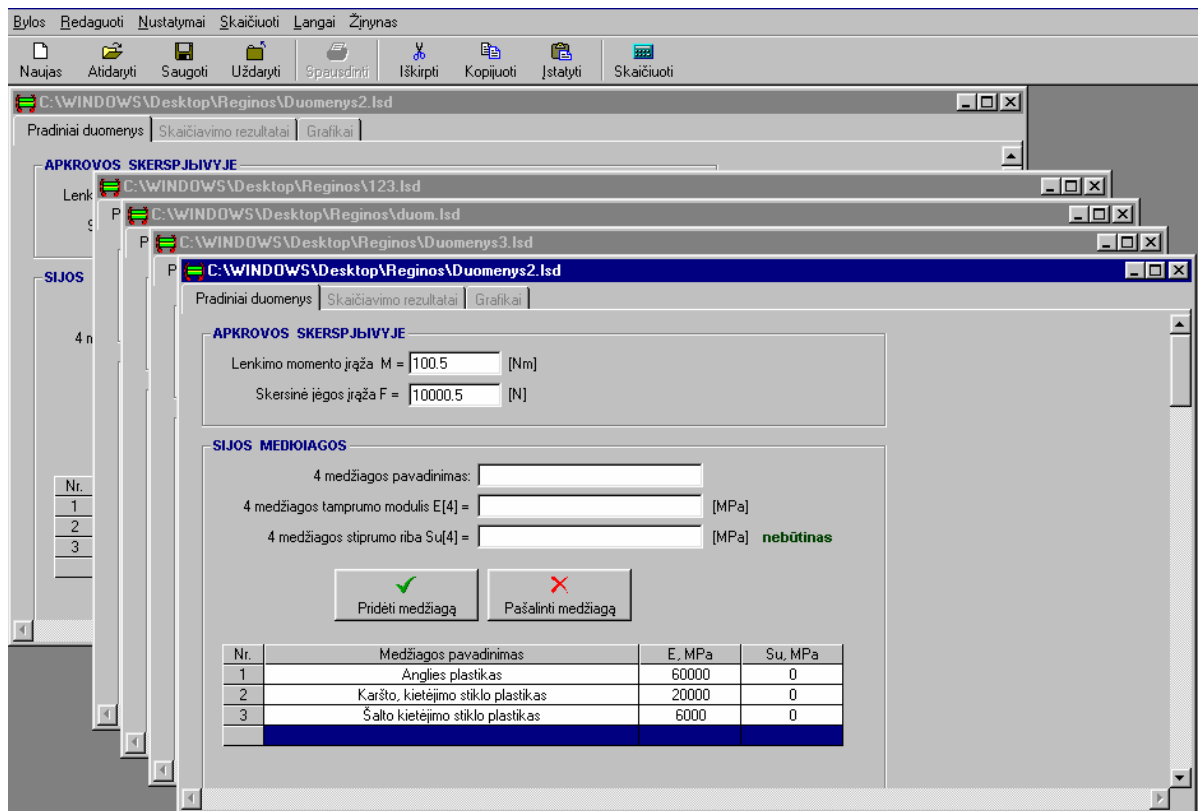
THE MODEL FOR DESIGN OF MULTI-LAYER CONSTRUCTION ELEMENTS SUBJECTED BY BENDING

Multi-layer construction elements improve technical characteristics of the product. There is exhaustive analysis of software in this work. Besides that the model for design of multi-layer construction elements subjected by bending is created in this work. The purpose of this program is calculation of stiffness and stresses for multi-layer constructions subjected by cross forces and bending moments.

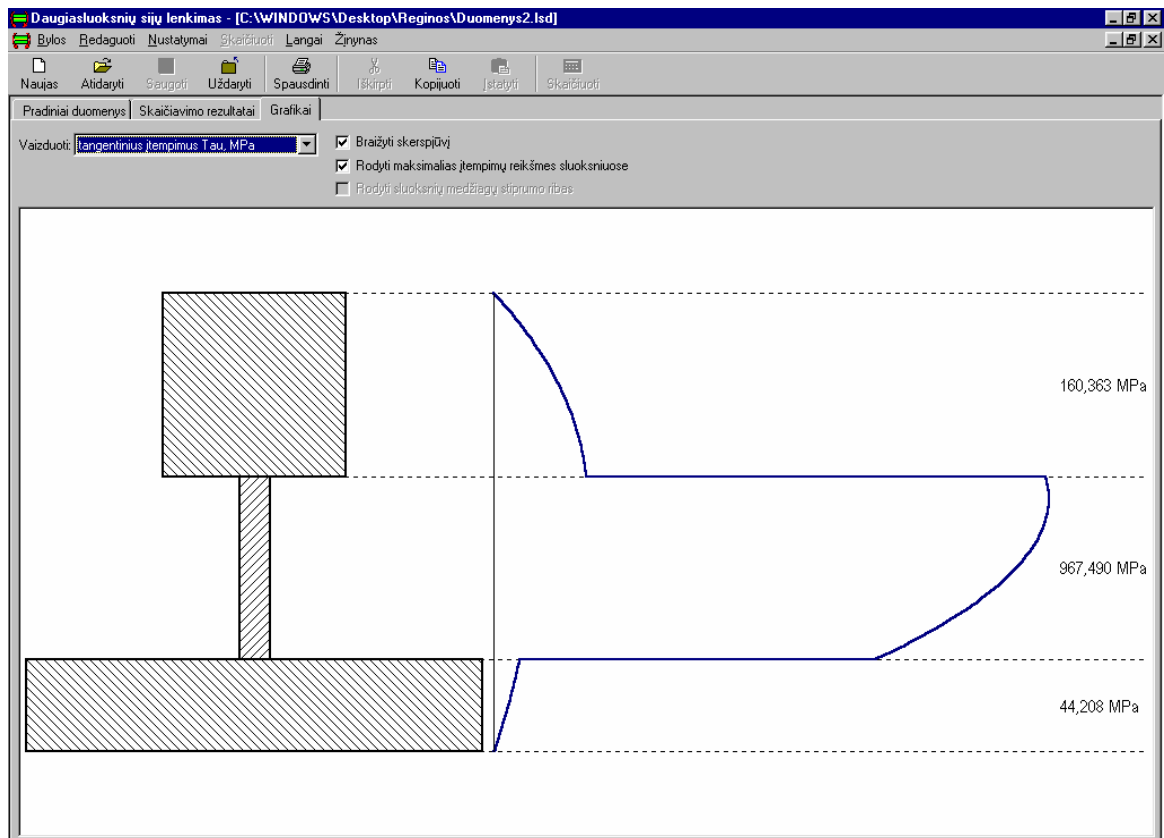
Modeling the design process of multi-layer construction elements is simple task that can be automatized by creation of modeling system. This program allows to calculate stiffness and strength of multi-layer construction elements subjected by cross forces and bending moments. It is possible to design various constructions of beams changing different parameters. The programme itself is simple for every student, engineer. This model can be used for scientific research, engineering calculations, laboratorial calculations.

The program allows to evaluate strength of multi-layer construction element subjected by bending. The construction will not hold out external load if intensity of stresses in one layer exceeds permissible stresses of strength limit of this layer material. The report is given in such case. The distribution of calculated stresses in cross-section of multi-layer construction elements can be shown graphically. It significantly facilitate designer's work.

PRIEDAI



14 pav. Langų išdėstymas kaskadomis



15 pav. Tangentinių įtempimų epiūra

Formos Pagrindas programa

```
Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal  
wMsg As Long, ByVal wParam As Long, ByVal lParam As Any) As Long  
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&, ByVal HelpFile$,  
ByVal wCommand%, dwData As Any)
```

```
Public YraDuomenys As Boolean
```

```
Public bi ' Sluoksniu plociai  
Public hi ' Sluoksniu auksciai  
Public Ai ' Sluoksniu skerspjuvio plotai  
Public Ii ' Sluoksniu ploto inercijos momentai  
Public ymi ' Sluoksniu skerspjuvio vidurio koordinate iki neutralines asies  
Public Si ' Sluoksniu ploto statiniai momentai  
Public Ei ' Sluoksniu tamprumo moduliai  
Public Bii ' Sluoksniu asiniai standumai  
Public Sui ' Sluoksniu medziagos stiprumo riba  
Public Sai ' Sluoksniu apacioje veikiantys normaliniai itempimai  
Public Svi ' Sluoksniu virsuje veikiantys normaliniai itempimai  
Public Tai ' Sluoksniu apacioje veikiantys tangentiniai itempimai  
Public Tvi ' Sluoksniu virsuje veikiantys tangentiniai itempimai
```

```
Public F As Double ' Tempimo jega  
Public M As Double ' Lenkimo momentas
```

```
Public B As Double ' Konstrukcijos asinis standumas  
Public D As Double ' Konstrukcijos standumas lenkimui  
Public yn As Double ' Konstrukcijos neutralios asies vieta  
Public Smax As Double ' Maksimali normalinio itempimo reiksme  
Public Simax As Double ' Maksimali itempimu intensyvumo reiksme  
Public Tmax As Double ' Maksimali tangentinio itempimo reiksme  
Public kiek As Integer ' Sluoksniu skaicius
```

```
Private Sub MDIForm_Activate()  
If ActiveForm Is Nothing Then  
Else  
ActiveForm.Refresh  
End If  
End Sub
```

```
Private Sub MDIForm_Load()  
On Error Resume Next  
ByluSk = 0  
Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)  
Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)  
Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)  
Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)  
Module1.Meniu_Mygtukai ByluSk  
End Sub
```

```
Private Sub MDIForm_Resize()  
On Error Resume Next  
If ActiveForm Is Nothing Then  
Else  
ActiveForm.Refresh  
End If  
End Sub
```

```
Private Sub MDIForm_Unload(Cancel As Integer)  
On Error Resume Next  
If Me.WindowState <> vbMinimized Then
```

```

        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
        SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

Private Sub mnuByloas_Naujas_Click()
    On Error Resume Next
    YraDuomenys = False
    Static IDocumentCount As Long
    Dim frmD As Langas
    IDocumentCount = IDocumentCount + 1
    Set frmD = New Langas
    frmD.Caption = "Bevardis" & IDocumentCount
    frmD.Show
    ByluSk = ByluSk + 1
    Module1.Meniu_Mygtukai ByluSk
End Sub

Private Sub mnuBylos_Atidaryti_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Atidaryti"
        .CancelError = False
        .Filter = "Duomenø bylos (*.lsd)|*.lsd|Visos bylos (*.*)|*.*"
        .ShowOpen
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
    End With
    mnuByloas_Naujas_Click
    ActiveForm.Caption = dlgCommonDialog.FileName
    gf = Nuskaityti_Pradinius_Duomenis(dlgCommonDialog.FileName)
    If gf = False Then
        pr = "Blogas failo " & dlgCommonDialog.FileName & " formatas."
        kas = MsgBox(pr, vbOKOnly + vbCritical, "Praneðimas")
        mnuBylos_Uzdaryti_Click
    End If
End Sub

Private Sub mnuBylos_Pabaiga_Click()
    On Error Resume Next
    Unload Me
End Sub

Private Sub mnuBylos_Saugoti_Click()
    On Error Resume Next
    If ActiveForm.SSTab1.Tab > 0 Then
        With dlgCommonDialog
            .DialogTitle = "Saugoti ..."
            .CancelError = False
            .Filter = "Skaièiavimo rezultatai (*.txt)|*.txt|Visos bylos (*.*)|*.*"
            .DefaultExt = ".txt"
            .ShowSave
            If Len(.FileName) = 0 Then Exit Sub
            ActiveForm.RichTextBox1.SaveFile .FileName, rtfText
        End With
    Else
        If Left$(ActiveForm.Caption, 8) = "Bevardis" Then
            With dlgCommonDialog
                .DialogTitle = "Saugoti ..."
                .CancelError = False
            End With
        End If
    End If
End Sub

```

```

        .Filter = "Duomenø bylos (*.lsd)|*.lsd|Visos bylos (*.*)|*.*"
        .DefaultExt = "lsd"
        .ShowSave
        If Len(.FileName) = 0 Then Exit Sub
        ActiveForm.Caption = .FileName
    End With
End If
    Saugoti_Pradinius_Duomenis ActiveForm.Caption
End If
End Sub

Private Sub mnuBylos_Saugoti_Vardu_Click()
    On Error Resume Next
    If ActiveForm.SSTab1.Tab > 0 Then
        mnuBylos_Saugoti_Click
        Exit Sub
    End If
    With dlgCommonDialog
        .DialogTitle = "Saugoti vardu ..."
        .CancelError = False
        .Filter = "Duomenø bylos (*.lsd)|*.lsd|Visos bylos (*.*)|*.*"
        .DefaultExt = "lsd"
        .ShowSave
        If Len(.FileName) = 0 Then Exit Sub
        ActiveForm.Caption = .FileName
    End With
    Saugoti_Pradinius_Duomenis ActiveForm.Caption
End Sub

Private Sub mnuBylos_Spausdinti_Click()
    On Error Resume Next
    With dlgCommonDialog
        .DialogTitle = "Spausdinti"
        .CancelError = True
        .Flags = cdIPDReturnDC + cdIPDNoPageNums
        If ActiveForm.rtfText.SelLength = 0 Then
            .Flags = .Flags + cdIPDAllPages
        Else
            .Flags = .Flags + cdIPDSelection
        End If
        .ShowPrinter
        If Err <> MSComDlg.cdlCancel Then
            Printer.Font.Name = "Times New Roman"
            Printer.Font.Charset = 0 "Baltic"
            Printer.Font.Size = 12
            Select Case ActiveForm.SSTab1.Tab
                Case 1: Printer.Print ActiveForm.RichTextBox1.Text
                Case 2: Select Case ActiveForm.Combo2.ListIndex
                    Case 0: kas = "Normaliniø átempimø "
                    Case 1: kas = "Tangentiniø átempimø "
                    Case 2: kas = "Átempimø intensyvumo "
                End Select
                kas = kas & "pasiskirstymas sijos skerspjuvyje"
                If ActiveForm.Check2.Value = 1 Then
                    kas = kas & Chr(10) & Chr(13) & _
                        "(skaièiai rodo maksimalias átempimø reikðmes sluoksniuose)"
                End If
                kas = kas & ":"
                Printer.Print kas
                Printer.PaintPicture ActiveForm.Picture3.Image, 0, 2 * Printer.TextHeight(kas)
            End Select
            Printer.EndDoc
        End If
    End Sub
End If

```

```

    End With
End Sub

Private Sub mnuBylos_Uzdaryti_Click()
    Unload Me.ActiveForm
    ByluSk = ByluSk - 1
    Module1.Meniu_Mygtukai ByluSk
End Sub

Private Sub mnuDarbas_Click()
    If Not Geri_Duomenys Then Exit Sub
    Pagrindiniai_Parametrai
    Rezultatu_Isvedimas
    YraDuomenys = True
    ActiveForm.Grafiko_Braizymas
    ActiveForm.SSTab1.Tab = 1
    ActiveForm.SSTab1.TabEnabled(1) = True
    ActiveForm.RichTextBox1.Visible = True
    ActiveForm.SSTab1.TabEnabled(2) = True
End Sub

Private Sub mnuLangai_Horizontaliai_Click()
    Me.Arrange vbTileHorizontal
End Sub

Private Sub mnuLangai_Kaskada_Click()
    Me.Arrange vbCascade
End Sub

Private Sub mnuLangai_Vertikaliai_Click()
    Me.Arrange vbTileVertical
End Sub

Private Sub mnuNustatymai_Irankiai_Click()
    mnuNustatymai_Irankiai.Checked = Not mnuNustatymai_Irankiai.Checked
    tbToolBar.Visible = mnuNustatymai_Irankiai.Checked
End Sub

Private Sub mnuNustatymai_Tekstas_Click()
    mnuNustatymai_Tekstas.Checked = Not mnuNustatymai_Tekstas.Checked
    For Each i In Me.Controls
        iTipas = TypeName(i)
        If iTipas = "ToolBar" Then
            For Each j In i.Buttons
                If mnuNustatymai_Tekstas.Checked Then
                    j.Caption = j.Key
                Else
                    j.Caption = ""
                End If
            Next
        End If
    Next
End Sub

Private Sub mnuPagalba_Apie_Click()
    On Error Resume Next
    Apie.Show vbModal, Me
End Sub

Private Sub mnuPagalba_Turinys_Click()
    On Error Resume Next
    htmlpav = App.Path & "\dsl.chm"
    kas = Shell("hh.exe " & htmlpav, vbNormalFocus)

```

```

End Sub

Private Sub mnuRedaguoti_Iskirpti_Click()
    On Error Resume Next
    ActiveForm.Iskirpti
End Sub

Private Sub mnuRedaguoti_Istatyti_Click()
    On Error Resume Next
    ActiveForm.Istatyti
End Sub

Private Sub mnuRedaguoti_Kopijuoti_Click()
    On Error Resume Next
    ActiveForm.Kopijuoti
End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
    On Error Resume Next
    Select Case Button.Key
        Case "Naujas"
            mnuByloas_Naujas_Click
        Case "Atidaryti"
            mnuBylos_Atidaryti_Click
        Case "Saugoti"
            mnuBylos_Saugoti_Click
        Case "Uzdaryti"
            mnuBylos_Uzdaryti_Click
        Case "Spausdinti"
            mnuBylos_Spauzdinti_Click
        Case "Iðkirpti"
            mnuRedaguoti_Iskirpti_Click
        Case "Kopijuoti"
            mnuRedaguoti_Kopijuoti_Click
        Case "Ástatyti"
            mnuRedaguoti_Istatyti_Click
        Case "Skaièiuoti"
            mnuDarbas_Click
    End Select
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If
End Sub

Private Sub mnuHelpContents_Click()
    Dim nRet As Integer
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation, Me.Caption
    Else
        On Error Resume Next

```

```

        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
    End If

End Sub

Private Sub Saugoti_Pradius_Duomenis(fv As String)
    On Error Resume Next
    Dim eile1 As Integer, eile2 As Integer, sk As Double
    eile1 = ActiveForm.MSHFlexGrid1.RowSel
    eile2 = ActiveForm.MSHFlexGrid2.RowSel
    Open fv For Output As #1
    ' Apkrovu saugojimas
    ActiveForm.Text1.Text = GerasSkaicius(ActiveForm.Text1.Text)
    sk = Val(ActiveForm.Text1.Text)
    Print #1, sk
    ActiveForm.Text2.Text = GerasSkaicius(ActiveForm.Text2.Text)
    sk = Val(ActiveForm.Text2.Text)
    Print #1, sk
    ' Medziagu saugojimas
    Print #1, ActiveForm.MSHFlexGrid1.Rows - 2
    For i = 1 To ActiveForm.MSHFlexGrid1.Rows - 2
        ActiveForm.MSHFlexGrid1.Row = i
        For j = 1 To 3
            ActiveForm.MSHFlexGrid1.Col = j
            If j = 1 Then
                Print #1, ActiveForm.MSHFlexGrid1.Text
            Else
                Print #1, GerasSkaicius(ActiveForm.MSHFlexGrid1.Text)
            End If
        Next
    Next
    ' Sluoksniu saugojimas
    Print #1, ActiveForm.MSHFlexGrid2.Rows - 2
    For i = 1 To ActiveForm.MSHFlexGrid2.Rows - 2
        ActiveForm.MSHFlexGrid2.Row = i
        For j = 1 To 2
            ActiveForm.MSHFlexGrid2.Col = j
            Print #1, GerasSkaicius(ActiveForm.MSHFlexGrid2.Text)
        Next
        ActiveForm.MSHFlexGrid2.Col = 3
        For j = 1 To ActiveForm.MSHFlexGrid1.Rows - 2
            ActiveForm.MSHFlexGrid1.Row = j
            ActiveForm.MSHFlexGrid1.Col = 1
            If ActiveForm.MSHFlexGrid1.Text = ActiveForm.MSHFlexGrid2.Text Then
                Print #1, j
                Exit For
            End If
        Next j
    Next
    Close #1
    ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid2, eile2
    ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid1, eile1
End Sub

Private Function Nuskaityti_Pradius_Duomenis(fv As String) As Boolean
    On Error Resume Next
    Open fv For Input As #1
    ' Apkrovu nuskaitymas
    Dim Esk As String, sk As Integer, i As Integer
    If EOF(1) Then

```

```

    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
Line Input #1, Esk
Esk = GerasSkaicius(Esk)
If Not ArSkaicius(Esk) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
ActiveForm.Text1.Text = Esk
If EOF(1) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
Line Input #1, Esk
Esk = GerasSkaicius(Esk)
If Not ArSkaicius(Esk) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
ActiveForm.Text2.Text = Esk
' Medziagu nuskaitymas
ActiveForm.Combo1.Clear
If EOF(1) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
Line Input #1, Esk
Esk = GerasSkaicius(Esk)
If Not ArSkaicius(Esk) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
sk = Val(Esk)
For i = 1 To sk
    ActiveForm.MSHFlexGrid1.Row = i
    ActiveForm.MSHFlexGrid1.Col = 0
    ActiveForm.MSHFlexGrid1.Text = i
    For j = 1 To 3
        ActiveForm.MSHFlexGrid1.Col = j
        If EOF(1) Then
            Close #1
            Nuskaityti_Pradinius_Duomenis = False
            Exit Function
        End If
        Line Input #1, Esk
        If j <> 1 Then
            Esk = GerasSkaicius(Esk)
            If Not ArSkaicius(Esk) Then
                Close #1
                Nuskaityti_Pradinius_Duomenis = False
                Exit Function
            End If
        End If
        ActiveForm.MSHFlexGrid1.Text = Esk
        If j = 1 Then
            ActiveForm.Combo1.AddItem ActiveForm.MSHFlexGrid1.Text

```

```

    End If
Next
For k = 0 To 3
    ActiveForm.MSHFlexGrid1.Col = k
    ActiveForm.MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
Next
ActiveForm.MSHFlexGrid1.Rows = ActiveForm.MSHFlexGrid1.Rows + 1
Next
' Sluoksniu nuskaitymas
If EOF(1) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
Line Input #1, Esk
Esk = GerasSkaicius(Esk)
If Not ArSkaicius(Esk) Then
    Close #1
    Nuskaityti_Pradinius_Duomenis = False
    Exit Function
End If
sk = Val(Esk)
For i = 1 To sk
    ActiveForm.MSHFlexGrid2.Row = i
    ActiveForm.MSHFlexGrid2.Col = 0
    ActiveForm.MSHFlexGrid2.Text = i
    For j = 1 To 2
        ActiveForm.MSHFlexGrid2.Col = j
        If EOF(1) Then
            Close #1
            Nuskaityti_Pradinius_Duomenis = False
            Exit Function
        End If
        Line Input #1, Esk
        Esk = GerasSkaicius(Esk)
        If Not ArSkaicius(Esk) Then
            Close #1
            Nuskaityti_Pradinius_Duomenis = False
            Exit Function
        End If
        ActiveForm.MSHFlexGrid2.Text = Esk
    Next
    If EOF(1) Then
        Close #1
        Nuskaityti_Pradinius_Duomenis = False
        Exit Function
    End If
    If EOF(1) Then
        Close #1
        Nuskaityti_Pradinius_Duomenis = False
        Exit Function
    End If
    Line Input #1, Esk
    Esk = GerasSkaicius(Esk)
    If Not ArSkaicius(Esk) Then
        Close #1
        Nuskaityti_Pradinius_Duomenis = False
        Exit Function
    End If
    ActiveForm.MSHFlexGrid1.Row = Esk
    ActiveForm.MSHFlexGrid1.Col = 1
    ActiveForm.MSHFlexGrid2.Col = 3
    ActiveForm.MSHFlexGrid2.Text = ActiveForm.MSHFlexGrid1.Text

```



```

For k = 0 To 3
    ActiveForm.MSHFlexGrid2.Col = k
    ActiveForm.MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
Next
ActiveForm.MSHFlexGrid2.Rows = ActiveForm.MSHFlexGrid2.Rows + 1
Next
ActiveForm.SluoksniuNumeravimas
ActiveForm.Skerspjuvis ActiveForm.Picture2
ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid1, ActiveForm.MSHFlexGrid1.Rows - 1
ActiveForm.MedzNr ActiveForm.MSHFlexGrid1.Rows - 1
ActiveForm.Combo1.ListIndex = 1
ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid2, ActiveForm.MSHFlexGrid2.Rows - 1
ActiveForm.SluoksnioNr ActiveForm.MSHFlexGrid2.Rows - 1
ActiveForm.Text1.SetFocus
Close #1
Nuskaityti_Pradinius_Duomenis = True
End Function

' Pagal medziagos pavadinima grazina jos standuma
Private Function MedzStandumas(pav As String) As Double
    Dim j As Integer
    Dim mp As String
    For j = 1 To ActiveForm.MSHFlexGrid1.Rows - 2
        ActiveForm.MSHFlexGrid1.Row = j
        ActiveForm.MSHFlexGrid1.Col = 1
        mp = ActiveForm.MSHFlexGrid1.Text
        If pav = mp Then
            ActiveForm.MSHFlexGrid1.Col = 2
            MedzStandumas = Val(ActiveForm.MSHFlexGrid1.Text)
            Exit Function
        End If
    Next
End Function

' Pagal medziagos pavadinima grazina jos stiprumo riba
Private Function MedzStiprumas(pav As String) As Double
    Dim j As Integer
    Dim mp As String
    For j = 1 To ActiveForm.MSHFlexGrid1.Rows - 2
        ActiveForm.MSHFlexGrid1.Row = j
        ActiveForm.MSHFlexGrid1.Col = 1
        mp = ActiveForm.MSHFlexGrid1.Text
        If pav = mp Then
            ActiveForm.MSHFlexGrid1.Col = 3
            MedzStiprumas = Val(ActiveForm.MSHFlexGrid1.Text)
            Exit Function
        End If
    Next
End Function

Private Sub Pradiniai_Duomenys()
    Dim eile1, eile2 As Integer
    eile1 = ActiveForm.MSHFlexGrid1.RowSel
    eile2 = ActiveForm.MSHFlexGrid2.RowSel
    M = Val(ActiveForm.Text1.Text) * 1000 ' Lenkimo momenta is [Nm] konvertuoju i [Nmm]
    F = Val(ActiveForm.Text2.Text)
    kiek = ActiveForm.MSHFlexGrid2.Rows - 3
    ReDim bi(kiek)
    ReDim hi(kiek)
    ReDim Ai(kiek)
    ReDim ymi(kiek)
    ReDim li(kiek)
    ReDim Si(kiek)

```

```

ReDim Ei(kiek)
ReDim Sui(kiek)
ReDim Bii(kiek)
ReDim Sai(kiek)
ReDim Svi(kiek)
ReDim Tai(kiek)
ReDim Tvi(kiek)
' Nuskaitau sluoksniu informacija
For j = 0 To kiek
    k = j + 1
    ActiveForm.MSHFlexGrid2.Row = k
    ActiveForm.MSHFlexGrid2.Col = 1
    bi(j) = Val(ActiveForm.MSHFlexGrid2.Text)
    ActiveForm.MSHFlexGrid2.Col = 2
    hi(j) = Val(ActiveForm.MSHFlexGrid2.Text)
    ActiveForm.MSHFlexGrid2.Col = 3
    Ei(j) = MedzStandumas(ActiveForm.MSHFlexGrid2.Text)
    Sui(j) = MedzStiprumas(ActiveForm.MSHFlexGrid2.Text)
Next
ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid1, Cint(eile1)
ActiveForm.IsrinktiEilute ActiveForm.MSHFlexGrid2, Cint(eile2)
End Sub

' Tikrina ar nurodytame TextBox'e ivestas tesingas skaicius.
' Jei taip, grazina True, priesingu atveju False ir parodo
' pranesima su irasu pr bei suaktyvina TextBox'a.
Private Function Yra_Skaicius(sk As TextBox, pr As String) As Boolean
    Yra_Skaicius = IsNumeric(sk.Text)
    If Not Yra_Skaicius Then
        X = MsgBox(pr, vbOKOnly + vbCritical, "Klaida")
        sk.SetFocus
        ActiveForm.TextGotFocus sk, 1
    End If
End Function

' Tikrina ar tinkami pradiniai duomenys.
' Jei tinkami grazina True priesingu atveju False
Private Function Geri_Duomenys() As Boolean
    Geri_Duomenys = Yra_Skaicius(ActiveForm.Text1, "Neteisingai avestas lenkimo momentas!" & Chr(13) &
    "Praðome pataisyti.")
    If Not Geri_Duomenys Then Exit Function
    Geri_Duomenys = Yra_Skaicius(ActiveForm.Text2, "Neteisingai avesta skersinë jëga!" & Chr(13) &
    "Praðome pataisyti.")
    If Not Geri_Duomenys Then Exit Function
    skM = Val(ActiveForm.Text1)
    skF = Val(ActiveForm.Text2)
    If skM = 0 And skF = 0 Then
        Geri_Duomenys = False
        X = MsgBox("Prieð skaiëiavimà reikia uþduoti apkrovà.", vbOKOnly + vbExclamation, "Praneðimas")
        Exit Function
    End If
    If ActiveForm.MSHFlexGrid2.Rows < 4 Then
        Geri_Duomenys = False
        X = MsgBox("Skaiëiavimo metodika reikalauja bent dviejø sluoksniø.", vbOKOnly + vbExclamation,
        "Praneðimas")
        Exit Function
    End If
End Function

Public Sub Pagrindiniai_Parametrai()
    Pradiniai_Duomenys
    ' Apskaiciuoju Ai, Bii, B ir neutralios asies vieta yn
    B = 0

```

```

sBihi = 0
For j = 0 To kiek
    Ai(j) = bi(j) * hi(j)
    Bii(j) = Ai(j) * Ei(j)
    B = B + Bii(j)
    sBihi = sBihi + Bii(j) * hi(j)
Next
sBi_shi = 0
dh = hi(0)
For j = 1 To kiek
    sBi_shi = sBi_shi + Bii(j) * dh
    dh = dh + hi(j)
Next
sBi_shi = 2 * sBi_shi
yn = (sBihi + sBi_shi) / 2# / B
' Apskaiciuojų sluoksniu ploto inercijos momentus Ii ir konstrukcijos standuma lenkimui D
dm = 0
D = 0
For j = 0 To kiek
    ymi(j) = hi(j) / 2# + dm - yn
    dm = dm + hi(j)
    Ii(j) = bi(j) * hi(j) ^ 3 / 12# + bi(j) * hi(j) * ymi(j) ^ 2
    D = D + Ei(j) * Ii(j)
Next
End Sub

Private Sub Rezultatu_Isvedimas()
    Dim endl As String, Text As String
    endl = Chr(13) & Chr(10)
    Text = ""
    ' Pradiniai duomenys
    Text = " PRADINIAI DUOMENYS" & endl & endl
    Text = Text & "Lenkimo momentas M = " & Format(M, "0.00") & " [N mm]" & endl
    Text = Text & "Skersinė jėga F = " & Format(F, "0.00") & " [N]" & endl & endl
    For j = 0 To kiek
        k = CStr(j + 1)
        Text = Text & k & "-o sluoksnio plotis b[" & k & "] = " & Format(bi(j), "0.00") & " [mm]" & endl
        Text = Text & k & "-o sluoksnio aukštis h[" & k & "] = " & Format(hi(j), "0.00") & " [mm]" & endl
        Text = Text & k & "-o sluoksnio medžiagos tamprumo modulis E[" & k & "] = " & Format(Ei(j), "0.00") &
" [MPa]" & endl & endl
    Next
    Text = Text & "Pastaba: ėia pirmas sluoksnis yra apatinis, paskutinis - viršutinis." & endl
    Text = Text & endl & endl & " SKAIĖIAVIMO REZULTATAI" & endl & endl
    Text = Text & "Konstrukcijos ađinis standumas B = " & Format(B, "0.000") & " [N]" & endl
    Text = Text & "Konstrukcijos standumas lenkimui D = " & Format(D, "0.000") & " [N mm^2]" & endl
    Text = Text & "Neutralės padėtis nuo skerspjūvio apaėios yn = " & Format(yn, "0.000") & " [mm]" & endl &
endl
    Text = Text & "Normaliniai átempimai S [MPa]:" & endl & endl
    Smax = 0#
    SmaxSluoksnis = "0"
    ' Normaliniu itempimu skaiciavimas ir isvedimas
    For j = 0 To kiek
        k = CStr(j + 1)
        dy1 = 0#
        For Z = 0 To j - 1
            dy1 = dy1 + hi(Z)
        Next
        Y1 = dy1 - yn
        Sai(j) = M * Ei(j) * Y1 / D
        If Abs(Smax) < Abs(Sai(j)) Then
            Smax = Sai(j)
            SmaxSluoksnis = k
        End If
    Next

```

```

Y2 = Y1 + hi(j)
Svi(j) = M * Ei(j) * Y2 / D
If Abs(Smax) < Abs(Svi(j)) Then
    Smax = Svi(j)
    SmaxSluoksnis = k
End If
Text = Text & "Sa[" & k & "] = " & Format(Sai(j), "0.000") & endl
Text = Text & "Sv[" & k & "] = " & Format(Svi(j), "0.000") & endl & endl
Next
' Tangentiniu itempimu skaiciavimas ir isvedimas
Tmax = 0
Text = Text & endl & "Tangentiniai átempimai T [MPa]:" & endl & endl
Tai(0) = 0
Text = Text & "Ta[1] = 0.000" & endl
Cx = 0
For j = 0 To kiek - 1
    k = CStr(j + 1)
    zz = CStr(j + 2)
    Cx = Cx + Ei(j) * Ai(j) * ymi(j)
    Tvi(j) = F * Abs(Cx) / D / bi(j)
    Text = Text & "Tv[" & k & "] = " & Format(Tvi(j), "0.000") & endl & endl
    If Abs(Tmax) < Abs(Tvi(j)) Then
        Tmax = Tvi(j)
        TmaxSluoksnis = j
    End If
    Tai(j + 1) = F * Abs(Cx) / D / bi(j + 1)
    Text = Text & "Ta[" & zz & "] = " & Format(Tai(j + 1), "0.000") & endl
    If Abs(Tmax) < Abs(Tai(j + 1)) Then
        Tmax = Tai(j + 1)
        TmaxSluoksnis = j + 1
    End If
Next
Tvi(kiek) = 0
Text = Text & "Tv[" & CStr(kiek + 1) & "] = 0.000" & endl & endl
' Tangentiniu itempimu skaiciavimas ties lenkimo neutrále:
' Sluoksnio, kuriame yra neutrále, nustatymas
NeutralesSluoksnis = 0
For j = 0 To kiek
    NeutralesSluoksnis = NeutralesSluoksnis + hi(j)
    If NeutralesSluoksnis >= yn Then
        NeutralesSluoksnis = j
        Exit For
    End If
Next
' Ekvivalentinio statinio momento nustatymas
Cx = 0
h1 = 0
For j = 0 To NeutralesSluoksnis - 1
    Cx = Cx + Ei(j) * Ai(j) * ymi(j)
    h1 = h1 + hi(j)
Next
hh = h1 - yn
Cx = Cx + Ei(NeutralesSluoksnis) * Abs(hh) * bi(NeutralesSluoksnis) * hh / 2#
t = F * Abs(Cx) / D / bi(j)
If Abs(Tmax) < Abs(t) Then
    Tmax = t
    TmaxSluoksnis = NeutralesSluoksnis
End If
' Maksimaliu itempimu intensyvumo nustatymas
Dim Simax As Double: Simax = 0
Dim SimaxSluoksnis As Integer
Text = Text & endl & "Átempimø intensyvumas Si [MPa]:" & endl & endl
For j = 0 To kiek

```

```

k = j + 1
S = (Sai(j) ^ 2 + 3# * Tai(j) ^ 2) ^ 0.5
Uzrasas = ""
If (Sui(j) > 0) And (S > Sui(j)) Then Uzrasas = " VIRĐYTA STIPRUMO RIBA !!!"
Text = Text & "Si_a[" & k & "]" = " & Format(S, "0.000") & Uzrasas & endl
If Abs(Simax) < Abs(S) Then
    Simax = S
    SimaxSluoksnis = j + 1
End If
S = (Svi(j) ^ 2 + 3# * Tvi(j) ^ 2) ^ 0.5
Uzrasas = ""
If (Sui(j) > 0) And (S > Sui(j)) Then Uzrasas = " VIRĐYTA STIPRUMO RIBA !!!"
Text = Text & "Si_v[" & k & "]" = " & Format(S, "0.000") & Uzrasas & endl & endl
If Abs(Simax) < Abs(S) Then
    Simax = S
    SimaxSluoksnis = j + 1
End If
Next
Text = Text & endl & "PASTABA: èai indeksas 'a' reiđkia apatinæ sluoksnio plokđtumà, o indeksas 'v' -
virđutinæ." & endl & endl
' Maksimaliu reiksmiu isvedimas
Text = Text & endl & "Maksimalios átempimø reikđmës" & endl & endl
Text = Text & "Smax = " & Format(Smax, "0.000") & " [MPa] - " & CStr(SmaxSluoksnis) & " sluoksnyje."
& endl
Text = Text & "Tmax = " & Format(Tmax, "0.000") & " [MPa] - " & CStr(TmaxSluoksnis + 1) & "
sluoksnyje." & endl & " "
Text = Text & "Simax = " & Format(Simax, "0.000") & " [MPa] - " & CStr(SimaxSluoksnis) & " sluoksnyje."
& endl & " "
ActiveForm.RichTextBox1.Text = Text
End Sub

```

Formos Langas programa

Public Top1, Top2, Top3, Left1 As Integer

```
Public Sub RedaguotiMedziaga()  
    Dim eileS As Integer, eileM As Integer  
    Text4.Text = Trim(Text4.Text)  
    If Text4.Text = "" Then  
        Pranesimas = "Nenurodytas medžiagos pavadinimas." & Chr(13) & "Prašome pataisyti."  
        kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Pranešimas")  
        Text4.SetFocus  
    Exit Sub  
    End If  
    If Not TestText(Text5, "Tamprumo modulis negali būti mažesnis arba lygus 0.") Then Exit Sub  
    Text7.Text = Trim(Text7.Text)  
    If Text7.Text = "" Then Text7.Text = 0  
    If Not TestText1(Text7, "Stiprumo riba modulis negali būti mažesnė už 0.") Then Exit Sub  
    eileM = MSHFlexGrid1.RowSel  
    MSHFlexGrid1.Col = 1  
    For i = 1 To MSHFlexGrid1.Rows - 2  
        MSHFlexGrid1.Row = i  
        If (MSHFlexGrid1.Text = Text4.Text) And (i <> eileM) Then  
            Pranesimas = "Medžiaga tokiu pavadinimu jau egzistuoja." & Chr(13) & "Prašome pataisyti."  
            kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Pranešimas")  
            Text4.SetFocus  
        Exit Sub  
    End If  
    Next i  
    eileS = MSHFlexGrid2.RowSel  
    ' Taisau medžiagu sarasa  
    MSHFlexGrid1.Row = eileM  
    MSHFlexGrid1.Col = 1  
    MSHFlexGrid1.Text = Text4.Text  
    MSHFlexGrid1.Col = 2  
    MSHFlexGrid1.Text = Text5.Text  
    MSHFlexGrid1.Col = 3  
    MSHFlexGrid1.Text = Text7.Text  
    MedzNr Str(MSHFlexGrid1.Rows - 1)  
    Medziagos  
    IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1  
    IsrinktiEilute MSHFlexGrid2, eileS  
    Text4.Text = ""  
    Text5.Text = ""  
    Text7.Text = ""  
    Text4.SetFocus  
End Sub
```

```
Public Sub RedaguotiSluoksni()  
    If Not TestText(Text6, "Sluoksniu plotis b negali būti mažesnis arba lygus 0.") Then  
        Text6.SetFocus  
    Exit Sub  
    End If  
    If Not TestText(Text3, "Sluoksniu aukštis h negali būti mažesnis arba lygus 0.") Then  
        Text3.SetFocus  
    Exit Sub  
    End If  
    Dim eile As Integer  
    eile = MSHFlexGrid2.RowSel  
    ' Taisau sluoksniu sarasa  
    MSHFlexGrid2.Row = eile  
    MSHFlexGrid2.Col = 1  
    MSHFlexGrid2.Text = Text6.Text  
    MSHFlexGrid2.Col = 2
```

```

MSHFlexGrid2.Text = Text3.Text
Skerspjuvis Picture2
Text6.Text = ""
Text3.Text = ""
IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1
Text6.SetFocus
End Sub

Public Sub IsrinktiEilute(kas As MSHFlexGrid, eNr As Integer)
    kas.Row = eNr
    kas.RowSel = eNr
    kas.Col = 0
    kas.ColSel = kas.Cols - 1
End Sub

Public Sub SluoksniuNumeravimas()
    MSHFlexGrid2.Col = 0
    For j = 1 To MSHFlexGrid2.Rows - 2
        MSHFlexGrid2.Row = j
        MSHFlexGrid2.Text = j
    Next
End Sub

Private Sub Check1_Click()
    Grafiko_Braizymas
End Sub

Private Sub Check2_Click()
    Grafiko_Braizymas
End Sub

Private Sub Check3_Click()
    If Check3.Value = Checked Then
        sp = RGB(196, 0, 0)
    Else
        sp = vbButtonText
    End If
    Check3.ForeColor = sp
    Grafiko_Braizymas
End Sub

Private Sub Combo1_Click()
    Dim eile As Integer
    eile = MSHFlexGrid2.RowSel
    If eile > MSHFlexGrid2.Rows - 2 Then
        Combo1.SetFocus
        Exit Sub
    End If
    ' Taisau sluoksniu sarasa
    MSHFlexGrid2.Row = eile
    MSHFlexGrid2.Col = 3
    MSHFlexGrid2.Text = Combo1.Text
    'Skerspjuvis Picture2
    IsrinktiEilute MSHFlexGrid2, eile
    Combo1.SetFocus
End Sub

Private Sub Combo2_Click()
    If Combo2.ListIndex = 2 Then
        Check3.Enabled = True
    Else
        Check3.Enabled = False
    End If

```

```
Grafiko_Braizymas  
End Sub
```

```
Private Sub Command5_Click()  
    ' Trinu nereikalinga sluoksni  
    If MSHFlexGrid2.RowSel <> (MSHFlexGrid2.Rows - 1) Then  
        Dim SNr As Integer: SNr = MSHFlexGrid2.RowSel  
        MSHFlexGrid2.RemoveItem MSHFlexGrid2.RowSel  
        MSHFlexGrid2.Col = 0  
        SluoksniuNumeravimas  
        MSHFlexGrid2.Row = MSHFlexGrid2.Rows - 1  
        MSHFlexGrid2.Text = ""  
        Skerspjuvis Picture2  
        IsrinktiEilute MSHFlexGrid2, SNr  
    End If  
End Sub
```

```
Private Sub Command6_Click()  
    If Not TestText(Text6, "Sluoksnio plotis b negali būti mažesnis arba lygus 0.") Then Exit Sub  
    If Not TestText(Text3, "Sluoksnio aukštis h negali būti mažesnis arba lygus 0.") Then Exit Sub  
    If Combo1.Text = "" Then  
        Pranesimas = "Praðome parinkti sluoksniu medþiagà."  
        kaip = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Praneðimas")  
        Combo1.SetFocus  
        Exit Sub  
    End If  
    eile = Val(MSHFlexGrid2.Rows) - 1  
    MSHFlexGrid2.Row = eile  
    MSHFlexGrid2.Col = 1  
    MSHFlexGrid2.Text = Text6.Text  
    MSHFlexGrid2.Col = 2  
    MSHFlexGrid2.Text = Text3.Text  
    MSHFlexGrid2.Col = 3  
    MSHFlexGrid2.Text = Combo1.Text  
    SluoksniuNr Str(MSHFlexGrid2.Rows)  
    For i = 0 To 3  
        MSHFlexGrid2.Col = i  
        MSHFlexGrid2.CellAlignment = flexAlignCenterCenter  
    Next  
    MSHFlexGrid2.Rows = MSHFlexGrid2.Rows + 1  
    Skerspjuvis Picture2  
    SluoksniuNumeravimas  
    IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1  
    Text6.Text = ""  
    Text3.Text = ""  
    Text6.SetFocus  
End Sub
```

```
Private Sub Form_Activate()  
    SStab_Meniu_Mygtukai  
    Select Case SStab1.Tab  
        Case 0  
            Skerspjuvis Picture2  
        Case 2  
            Grafiko_Braizymas  
    End Select  
End Sub
```

```
Private Sub Form_GotFocus()  
    'fMainForm.YraDuomenys = True  
    Select Case SStab1.Tab  
        Case 0  
            Skerspjuvis Picture2
```



```

Case 2
  Grafiko_Braizymas
End Select
End Sub

```

```

Private Sub Form_Load()
  'Form_Resize
  RichTextBox1.Visible = False
  SSTab1.TabEnabled(1) = False
  SSTab1.TabEnabled(2) = False
  ' Grafikai
  Combo2.Clear
  Combo2.AddItem "normalinius átempinus Sigma, MPa"
  Combo2.AddItem "tangentinius átempimus Tau, MPa"
  Combo2.AddItem "átempimø intensyvumas Sigma_i, MPa"
  Combo2.ListIndex = 0
  ' Pradiniai duomenys
  SSTab1.Tab = 0
  MSHFlexGrid1.Row = 0
  MSHFlexGrid1.Col = 0
  MSHFlexGrid1.Text = "Nr."
  MSHFlexGrid1.Col = 1
  MSHFlexGrid1.Text = "Medþiagos pavadinimas"
  MSHFlexGrid1.Col = 2
  MSHFlexGrid1.Text = "E, MPa"
  MSHFlexGrid1.Col = 3
  MSHFlexGrid1.Text = "Su, MPa"
  For i = 0 To 3
    MSHFlexGrid1.Col = i
    MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
  Next
  MSHFlexGrid1.Row = 1 ' Padarome aktyvia 1 eilute
  MSHFlexGrid1.ColWidth(0) = 500
  MSHFlexGrid1.ColWidth(1) = 4410
  MSHFlexGrid1.ColWidth(2) = 1200
  MSHFlexGrid1.ColWidth(3) = 1200
  IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1

  Combo1.Clear

  MSHFlexGrid2.Row = 0
  MSHFlexGrid2.Col = 0
  MSHFlexGrid2.Text = "Nr."
  MSHFlexGrid2.Col = 1
  MSHFlexGrid2.Text = "b, mm"
  MSHFlexGrid2.Col = 2
  MSHFlexGrid2.Text = "h, mm"
  MSHFlexGrid2.Col = 3
  MSHFlexGrid2.Text = "Medþiaga"
  For i = 0 To 3
    MSHFlexGrid2.Col = i
    MSHFlexGrid2.CellAlignment = flexAlignCenterCenter
  Next
  MSHFlexGrid2.Row = 1 ' Padarome aktyvia 1 eilute
  MSHFlexGrid2.ColWidth(0) = 500
  MSHFlexGrid2.ColWidth(1) = 1200
  MSHFlexGrid2.ColWidth(2) = 1200
  MSHFlexGrid2.ColWidth(3) = 3960
  MSHFlexGrid1.RowHeightMin = MSHFlexGrid1.CellHeight
  MSHFlexGrid2.RowHeightMin = MSHFlexGrid2.CellHeight
  IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.Rows - 1

  Top1 = Frame1.Top

```

```

Top2 = Frame2.Top
Top3 = Frame3.Top
Left1 = Frame1.Left

RodytiSkerspjuvi = True
End Sub

Public Sub MedzNr(Nr As Integer)
Label7.Caption = Nr & " medžiagos pavadinimas: "
Label8.Caption = Nr & " medžiagos tamprumo modulis E[" & Nr & "] = "
Label15.Caption = Nr & " medžiagos stiprumo riba Su[" & Nr & "] = "
End Sub

Public Sub SluoksnioNr(Nr As Integer)
Label10.Caption = Nr & " sluoksnio plotis b[" & Nr & "] = "
Label6.Caption = Nr & " sluoksnio aukštis h[" & Nr & "] = "
Label11.Caption = Nr & " sluoksnio medžiaga: "
End Sub

Private Sub Medziagos()
Combo1.Clear
MSHFlexGrid1.Col = 1
Dim i As Integer
For i = 1 To MSHFlexGrid1.Rows - 2
MSHFlexGrid1.Row = i
Combo1.AddItem MSHFlexGrid1.Text
Next
If Combo1.ListIndex = -1 Then Combo1.ListIndex = 0
End Sub

' Tikrina ar skaičius ne mažesnis ir nelygus 0.
Private Function TestText(txt As TextBox, Pranes As String) As Boolean
Dim YraSkaicius As Boolean, Ilgis As Integer, Geras As String, Skaicius As String
txt.Text = GerasSkaicius(txt.Text)
YraSkaicius = ArSkaicius(txt.Text)
sk = Val(txt.Text)
If (sk <= 0) Or (YraSkaicius = False) Then
Pranesimas = Pranes & Chr(13) & "Praðome pataisyti."
kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Praneðimas")
txt.SetFocus
TestText = False
Exit Function
End If
TestText = True
End Function

' Tikrina ar skaičius ne mažesnis uz 0.
Private Function TestText1(txt As TextBox, Pranes As String) As Boolean
Dim YraSkaicius As Boolean, Ilgis As Integer, Geras As String, Skaicius As String
txt.Text = GerasSkaicius(txt.Text)
YraSkaicius = ArSkaicius(txt.Text)
sk = Val(txt.Text)
If (sk < 0) Or (YraSkaicius = False) Then
Pranesimas = Pranes & Chr(13) & "Praðome pataisyti."
kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Praneðimas")
txt.SetFocus
TestText1 = False
Exit Function
End If
TestText1 = True
End Function

Private Sub Command1_Click()

```

```

Text4.Text = Trim(Text4.Text)
If Text4.Text = "" Then
    Pranesimas = "Nenurodytas medžiagos pavadinimas." & Chr(13) & "Praðome pataisyti."
    kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Praneðimas")
    Text4.SetFocus
    Exit Sub
End If
If Not TestText(Text5, "Tamprumo modulis negali bûti maþesnis arba lygus 0.") Then Exit Sub
Text7.Text = Trim(Text7.Text)
If Text7.Text = "" Then Text7.Text = 0
If Not TestText1(Text7, "Stiprumo riba modulis negali bûti maþesnë uþ 0.") Then Exit Sub
MSHFlexGrid1.Col = 1
For i = 1 To MSHFlexGrid1.Rows - 2
    MSHFlexGrid1.Row = i
    If MSHFlexGrid1.Text = Text4.Text Then
        Pranesimas = "Medþiaga tokiu pavadinimu jau egzistuoja." & Chr(13) & "Praðome pataisyti."
        kas = MsgBox(Pranesimas, vbOKOnly + vbExclamation, "Praneðimas")
        Text4.SetFocus
        Exit Sub
    End If
Next i
eile = Val(MSHFlexGrid1.Rows) - 1
MSHFlexGrid1.Row = eile
MSHFlexGrid1.Col = 0
MSHFlexGrid1.Text = eile
MSHFlexGrid1.Col = 1
MSHFlexGrid1.Text = Text4.Text
MSHFlexGrid1.Col = 2
MSHFlexGrid1.Text = Text5.Text
MSHFlexGrid1.Col = 3
MSHFlexGrid1.Text = Text7.Text
MedzNr Str(MSHFlexGrid1.Rows)
For i = 0 To 3
    MSHFlexGrid1.Col = i
    MSHFlexGrid1.CellAlignment = flexAlignCenterCenter
Next
MSHFlexGrid1.Rows = MSHFlexGrid1.Rows + 1
Medziagos
IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.Rows - 1
Text4.Text = ""
Text5.Text = ""
Text7.Text = ""
Text4.SetFocus
End Sub

Private Sub Command2_Click()
    ' Tikrinu ar medziaga nera naudojama
    Dim i As Integer
    Dim NaudojamaMedz As Boolean
    NaudojamaMedz = False
    MSHFlexGrid2.Col = 3
    MSHFlexGrid1.Col = 1
    MSHFlexGrid1.Row = MSHFlexGrid1.RowSel
    mp = MSHFlexGrid1.Text
    For i = 1 To MSHFlexGrid2.Rows - 2
        MSHFlexGrid2.Row = i
        If MSHFlexGrid2.Text = mp Then
            NaudojamaMedz = True
            Exit For
        End If
    Next
    If NaudojamaMedz Then

```

```

    X = MsgBox("Medžiaga "" & MSHFlexGrid1.Text & "" yra naudojama.", vbOKOnly + vbExclamation,
"Pranešimas")
    Exit Sub
End If
' Trinus nereikalinga medžiaga
Dim MNr As Integer: MNr = MSHFlexGrid1.RowSel
If MSHFlexGrid1.RowSel <> (MSHFlexGrid1.Rows - 1) Then
    MSHFlexGrid1.RemoveItem MSHFlexGrid1.RowSel
End If
MSHFlexGrid1.Col = 0
kiek = MSHFlexGrid1.Rows - 2
For i = 1 To kiek
    MSHFlexGrid1.Row = i
    MSHFlexGrid1.Text = i
Next
Medžiagos
IsrinktiEilute MSHFlexGrid1, MNr
End Sub

Public Sub TextGotFocus(ByVal txt As TextBox, Optional kaip As Integer = 0)
    If kaip <> 0 Then
        txt.Select = 0
        txt.SelectLength = Len(txt.Text)
    End If
    Dim t
    For Each t In Me.Controls
        If TypeOf t Is TextBox Then t.Tag = 0
    Next t
    txt.Tag = 1
End Sub

Private Sub Form_Resize()
    On Error Resume Next
    ' Skaiciavimo rezultatai
    RichTextBox1.Left = 60
    RichTextBox1.Top = 360
    RichTextBox1.Width = Me.ScaleWidth - 2 * RichTextBox1.Left
    RichTextBox1.Height = Me.ScaleHeight - RichTextBox1.Top - RichTextBox1.Left
    ' Grafikai
    Picture3.Left = 100
    Picture3.Top = Check3.Top + Check3.Height + 100
    Picture3.Width = Me.ScaleWidth - 2 * 100
    Picture3.Height = Me.ScaleHeight - Picture3.Top - 100
    ' Pradiniai duomenys
    SStab1.Width = Me.ScaleWidth
    SStab1.Height = Me.ScaleHeight
    Picture1.Width = Me.ScaleWidth
    Picture1.Height = Me.ScaleHeight - Picture1.Top
    VScroll1.Height = Picture1.ScaleHeight - HScroll1.Height
    VScroll1.Left = Picture1.ScaleWidth - VScroll1.Width
    vkiek = Frame3.Top + Frame3.Height + HScroll1.Height - Picture1.ScaleHeight
    VScroll1.Max = vkiek + Frame3.Left
    If vkiek <= 0 Then
        VScroll1.Enabled = False
    Else
        VScroll1.Enabled = True
    End If
    HScroll1.Width = Picture1.ScaleWidth - VScroll1.Width
    HScroll1.Top = Picture1.ScaleHeight - HScroll1.Height
    hkiek = Frame1.Left + Frame1.Width + VScroll1.Width - Picture1.ScaleWidth
    HScroll1.Max = hkiek + Frame3.Left
    If hkiek <= 0 Then
        HScroll1.Enabled = False
    End If
End Sub

```

```

Else
    HScroll1.Enabled = True
End If
Select Case SSTab1.Tab
    Case 0
        Skerspjuvis Picture2
    Case 2
        Grafiko_Braizymas
End Select
End Sub

Private Sub HerizScrolling()
    Frame1.Left = Left1 - HScroll1.Value
    Frame2.Left = Frame1.Left
    Frame3.Left = Frame1.Left
    Skerspjuvis Picture2
End Sub

Private Sub Form_Unload(Cancel As Integer)
    Module1.ByluSk = Module1.ByluSk - 1
    Module1.Meniu_Mygtukai Module1.ByluSk
End Sub

Private Sub HScroll1_Change()
    HerizScrolling
End Sub

Private Sub HScroll1_Scroll()
    HerizScrolling
End Sub

Private Sub MSHFlexGrid1_Click()
    MSHFlexGrid1.Row = MSHFlexGrid1.RowSel
    MSHFlexGrid1.Col = 1
    Text4.Text = MSHFlexGrid1.Text
    MSHFlexGrid1.Col = 2
    Text5.Text = MSHFlexGrid1.Text
    MSHFlexGrid1.Col = 3
    Text7.Text = MSHFlexGrid1.Text
    MedzNr MSHFlexGrid1.Row
    IsrinktiEilute MSHFlexGrid1, MSHFlexGrid1.RowSel
End Sub

Private Sub MSHFlexGrid1_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyDelete And Shift = 0 Then
        Command2_Click
    End If
End Sub

Private Sub MSHFlexGrid2_Click()
    MSHFlexGrid2.Row = MSHFlexGrid2.RowSel
    MSHFlexGrid2.Col = 1
    Text6.Text = MSHFlexGrid2.Text
    MSHFlexGrid2.Col = 2
    Text3.Text = MSHFlexGrid2.Text
    MSHFlexGrid2.Col = 3
    Dim i As Integer
    If MSHFlexGrid2.Row <> MSHFlexGrid2.Rows - 1 Then
        For i = 0 To Combo1.ListCount
            If Combo1.List(i) = MSHFlexGrid2.Text Then
                Combo1.ListIndex = i
            Exit For
        End If
    End If
End Sub

```

```

    Next
End If
SluoksnioNr MSHFlexGrid2.Row
IsrinktiEilute MSHFlexGrid2, MSHFlexGrid2.RowSel
End Sub

Private Sub MSHFlexGrid2_KeyDown(KeyCode As Integer, Shift As Integer)
    If KeyCode = vbKeyDelete And Shift = 0 Then
        Command5_Click
    End If
End Sub

Public Sub SSTab_Meniu_Mygtukai()
    Select Case SSTab1.Tab
        Case 0
            fMainForm.mnuBylos_Saugoti = True
            fMainForm.mnuBylos_Saugoti_Vardu.Enabled = True
            fMainForm.mnuBylos_Spausdinti.Enabled = False
            fMainForm.mnuRedaguoti_Iskirpti = True
            fMainForm.mnuRedaguoti_Istatyti = True
            fMainForm.mnuDarbas.Enabled = True
            fMainForm.tbToolBar.Buttons.Item(3).Enabled = True ' Saugoti
            fMainForm.tbToolBar.Buttons.Item(8).Enabled = True ' Iskirpti
            fMainForm.tbToolBar.Buttons.Item(10).Enabled = True ' Istatyti
            fMainForm.tbToolBar.Buttons.Item(6).Enabled = False ' Spausdinti
            fMainForm.tbToolBar.Buttons.Item(12).Enabled = True ' Skaiciuoti
        Case 1
            fMainForm.mnuBylos_Saugoti = True
            fMainForm.mnuBylos_Saugoti_Vardu.Enabled = True
            fMainForm.mnuBylos_Spausdinti.Enabled = True
            fMainForm.mnuRedaguoti_Iskirpti = False
            fMainForm.mnuRedaguoti_Istatyti = False
            fMainForm.mnuDarbas.Enabled = False
            fMainForm.tbToolBar.Buttons.Item(3).Enabled = True ' Saugoti
            fMainForm.tbToolBar.Buttons.Item(8).Enabled = False ' Iskirpti
            fMainForm.tbToolBar.Buttons.Item(10).Enabled = False ' Istatyti
            fMainForm.tbToolBar.Buttons.Item(6).Enabled = True ' Spausdinti
            fMainForm.tbToolBar.Buttons.Item(12).Enabled = False ' Skaiciuoti
        Case 2
            fMainForm.mnuBylos_Saugoti = False
            fMainForm.mnuBylos_Saugoti_Vardu.Enabled = False
            fMainForm.mnuBylos_Spausdinti.Enabled = True
            fMainForm.mnuRedaguoti_Iskirpti = False
            fMainForm.mnuRedaguoti_Istatyti = False
            fMainForm.mnuDarbas.Enabled = False
            fMainForm.tbToolBar.Buttons.Item(3).Enabled = False ' Saugoti
            fMainForm.tbToolBar.Buttons.Item(8).Enabled = False ' Iskirpti
            fMainForm.tbToolBar.Buttons.Item(10).Enabled = False ' Istatyti
            fMainForm.tbToolBar.Buttons.Item(6).Enabled = True ' Spausdinti
            fMainForm.tbToolBar.Buttons.Item(12).Enabled = False ' Skaiciuoti
    End Select
End Sub

Private Sub SSTab1_Click(PreviousTab As Integer)
    SSTab_Meniu_Mygtukai
    Select Case SSTab1.Tab
        Case 0
            RichTextBox1.Visible = False
            Picture3.Visible = False
            Skerspjuvis Picture2
        Case 1
            RichTextBox1.Visible = True
            Picture3.Visible = False
    End Select
End Sub

```

```

Case 2
    RichTextBox1.Visible = False
    Picture3.Visible = True
    Grafiko_Braizymas
End Select
End Sub

Private Sub Text1_GotFocus()
    TextGotFocus Text1
End Sub

Private Sub Text2_GotFocus()
    TextGotFocus Text2
End Sub

Private Sub Text3_GotFocus()
    TextGotFocus Text3
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If MSHFlexGrid2.RowSel > MSHFlexGrid2.Rows - 2 Then
            Command6_Click
        Else
            RedaguotiSluoksni
        End If
    End If
End Sub

Private Sub Text4_GotFocus()
    TextGotFocus Text4
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If MSHFlexGrid1.RowSel > MSHFlexGrid1.Rows - 2 Then
            Command1_Click
        Else
            RedaguotiMedziaga
        End If
    End If
End Sub

Private Sub Text5_GotFocus()
    TextGotFocus Text5
End Sub

Private Sub VerticalScrolling()
    Frame1.Top = Top1 - VScroll1.Value
    Frame2.Top = Top2 - VScroll1.Value
    Frame3.Top = Top3 - VScroll1.Value
    Skerspjuvis Picture2
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If MSHFlexGrid1.RowSel > MSHFlexGrid1.Rows - 2 Then
            Command1_Click
        Else
            RedaguotiMedziaga
        End If
    End If
End Sub

```

```

Private Sub Text6_GotFocus()
    TextGotFocus Text6
End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If MSHFlexGrid2.RowSel > MSHFlexGrid2.Rows - 2 Then
            Command6_Click
        Else
            RedaguotiSluoksni
        End If
    End If
End Sub

Private Sub Text7_KeyPress(KeyAscii As Integer)
    If KeyAscii = 13 Then
        If MSHFlexGrid1.RowSel > MSHFlexGrid1.Rows - 2 Then
            Command1_Click
        Else
            RedaguotiMedziaga
        End If
    End If
End Sub

Private Sub SukeistiVietomis(Nr1 As Integer, Nr2 As Integer)
    If Nr1 > 0 And Nr1 < MSHFlexGrid2.Rows - 1 And Nr2 > 0 And Nr2 < MSHFlexGrid2.Rows - 1 Then
        MSHFlexGrid2.Row = Nr1
        MSHFlexGrid2.Col = 1: bb1 = MSHFlexGrid2.Text
        MSHFlexGrid2.Col = 2: hh1 = MSHFlexGrid2.Text
        MSHFlexGrid2.Col = 3: mm1 = MSHFlexGrid2.Text
        MSHFlexGrid2.Row = Nr2
        MSHFlexGrid2.Col = 1: bb2 = MSHFlexGrid2.Text
        MSHFlexGrid2.Col = 2: hh2 = MSHFlexGrid2.Text
        MSHFlexGrid2.Col = 3: mm2 = MSHFlexGrid2.Text
        MSHFlexGrid2.Row = Nr1
        MSHFlexGrid2.Col = 1: MSHFlexGrid2.Text = bb2
        MSHFlexGrid2.Col = 2: MSHFlexGrid2.Text = hh2
        MSHFlexGrid2.Col = 3: MSHFlexGrid2.Text = mm2
        MSHFlexGrid2.Row = Nr2
        MSHFlexGrid2.Col = 1: MSHFlexGrid2.Text = bb1
        MSHFlexGrid2.Col = 2: MSHFlexGrid2.Text = hh1
        MSHFlexGrid2.Col = 3: MSHFlexGrid2.Text = mm1
        IsrinktiEilute MSHFlexGrid2, Nr2
    End If
End Sub

Private Sub UpDown1_DownClick()
    SukeistiVietomis MSHFlexGrid2.RowSel, MSHFlexGrid2.RowSel + 1
    Skerspjuvis Picture2
End Sub

Private Sub UpDown1_UpClick()
    SukeistiVietomis MSHFlexGrid2.RowSel, MSHFlexGrid2.RowSel - 1
    Skerspjuvis Picture2
End Sub

Private Sub VScroll1_Scroll()
    VerticalScrolling
End Sub

Private Sub VScroll1_Change()
    VerticalScrolling

```



```

End Sub

Private Sub Pranesimas_Text(kas As TextBox, txt As String)
    xxx = MsgBox(txt & Chr(13) & Chr(10) & "Praðome pataisyti.", vbOKOnly + vbExclamation, "Praneðimas")
    kas.SetFocus
    kas.SelStart = 0
    kas.SelLength = Len(kas.Text)
End Sub

Public Sub Grafiko_Braizymas()
    If fMainForm.YraDuomenys <> True Then Exit Sub
    Dim TaskuSkSluoksnyje As Integer: TaskuSkSluoksnyje = 50
    Dim Y1, kH, Daugiklis, Mastelis, sMastelis, Plotis, bmax, hmax, Tikslumas As Double
    Tikslumas = 0.0000000001
    Dim Kraskai As Integer
    Dim Sigma, h As Variant
    Krastai = 75
    ' Istrinu buvusi vaizda
    Picture3.Cls
    ' Perskaiciuoju pagrindinius parametrus
    fMainForm.Pagrindiniai_Parametrai
    ' Nustatau maksimalu skerspjuvio ploti ir auksti
    If MSHFlexGrid2.Rows < 3 Then Exit Sub
    hmax = 0
    bmax = 0
    For j = 1 To MSHFlexGrid2.Rows - 1
        MSHFlexGrid2.Row = j
        MSHFlexGrid2.Col = 1
        bh = Val(MSHFlexGrid2.Text)
        If bmax < bh Then bmax = bh
        MSHFlexGrid2.Col = 2
        bh = Val(MSHFlexGrid2.Text)
        hmax = hmax + bh
    Next
    ' Apskaiciuoju itempimu reiksmes sluoksniuose
    Dim dyy, dyy1 As Double
    Select Case Combo2.ListIndex
        Case 0 ' Normaliniai itempimai
            If fMainForm.M = 0 Then
                kiek = 0
                Smax = 1
                Label19.Visible = True
                Label19.Caption = "Normaliniø átempimø nėra"
            Else
                kiek = 2 * (fMainForm.kiek + 2)
                ReDim Sigma(kiek)
                ReDim h(kiek)
                Smax = 0
                Smin = 0
                Sigma(0) = 0#
                h(0) = 0
                i = 1
                For j = 0 To fMainForm.kiek
                    dy1 = 0#
                    For zz = 0 To j - 1
                        dy1 = dy1 + fMainForm.hi(zz)
                    Next
                    Y1 = dy1 - fMainForm.yn
                    Sigma(i) = fMainForm.M * fMainForm.Ei(j) * Y1 / fMainForm.D
                    h(i) = h(i - 1)
                    If Smax < Sigma(i) Then Smax = Sigma(i)
                    If Smin > Sigma(i) Then Smin = Sigma(i)
                    i = i + 1
                Next
            End Select
    End Sub

```

```

Y2 = Y1 + fMainForm.hi(j)
Sigma(i) = fMainForm.M * fMainForm.Ei(j) * Y2 / fMainForm.D
h(i) = h(i - 1) + fMainForm.hi(j)
If Smax < Sigma(i) Then Smax = Sigma(i)
If Smin > Sigma(i) Then Smin = Sigma(i)
i = i + 1
Next
Sigma(i) = 0
h(i) = hmax
Label19.Visible = False
End If
Case 1 ' Tangentiniai itempimai
If fMainForm.F = 0 Then
kiek = 0
Smax = 1
Label19.Visible = True
Label19.Caption = "Tangentiniø átempimø nëra"
Else
kiek = TaskuSkSluoksnyje * (fMainForm.kiek + 1) + 2
ReDim Sigma(kiek)
ReDim h(kiek)
Sigma(0) = 0
h(0) = 0
Smax = 0
Smin = 0
Cx = 0 ' Ploto statinis momentas
Hsum = 0 ' Suminis aukstis
yy = 0 ' y koordinate
i = 1
For j = 0 To fMainForm.kiek
If (Hsum < fMainForm.yn) And ((Hsum + fMainForm.hi(j)) > fMainForm.yn) Then
sk1 = CInt(TaskuSkSluoksnyje * (fMainForm.yn - Hsum) / fMainForm.hi(j)) + 1
sk2 = TaskuSkSluoksnyje - sk1 - 1
dyy = (fMainForm.yn - Hsum) / sk1
dyy1 = Abs(fMainForm.hi(j) - fMainForm.yn + Hsum) / sk2
Else
dyy = fMainForm.hi(j) / (TaskuSkSluoksnyje - 1)
dyy1 = -1
End If
hh = 0
For zz = 0 To (TaskuSkSluoksnyje - 1)
Cx1 = Cx + fMainForm.Ei(j) * (fMainForm.bi(j) * hh) * (Hsum + hh / 2 - fMainForm.yn)
Sigma(i) = fMainForm.F * Abs(Cx1) / fMainForm.D / fMainForm.bi(j)
h(i) = yy
If Smax < Sigma(i) Then Smax = Sigma(i)
If Smin > Sigma(i) Then Smin = Sigma(i)
i = i + 1
If (zz > sk1 - 1) And (dyy1 <> -1) Then dyy = dyy1
If zz < (TaskuSkSluoksnyje - 1) Then yy = yy + dyy
hh = hh + dyy
Next zz
Cx = Cx + fMainForm.Ei(j) * fMainForm.Ai(j) * fMainForm.ymi(j)
Hsum = Hsum + fMainForm.hi(j)
Next j
Sigma(i) = 0
h(i) = hmax
Label19.Visible = False
End If
Case 2 ' Itempimu intensyvumas
kiek = TaskuSkSluoksnyje * (fMainForm.kiek + 1) + 2
ReDim Sigma(kiek)
ReDim h(kiek)
Sigma(0) = 0

```

```

h(0) = 0
Smax = 0
Smin = 0
Cx = 0 ' Ploto statinis momentas
Hsum = 0 ' Suminis aukstis
yy = 0 ' y koordinate
i = 1
For j = 0 To fMainForm.kiek
  If (Hsum < fMainForm.yn) And ((Hsum + fMainForm.hi(j)) > fMainForm.yn) Then
    sk1 = CInt(TaskuSkSluoksnyje * (fMainForm.yn - Hsum) / fMainForm.hi(j)) + 1
    sk2 = TaskuSkSluoksnyje - sk1 - 1
    dyy = (fMainForm.yn - Hsum) / sk1
    dyy1 = Abs(fMainForm.hi(j) - fMainForm.yn + Hsum) / sk2
  Else
    dyy = fMainForm.hi(j) / (TaskuSkSluoksnyje - 1)
    dyy1 = -1
  End If
  hh = 0
  For zz = 0 To (TaskuSkSluoksnyje - 1)
    Cx1 = Cx + fMainForm.Ei(j) * (fMainForm.bi(j) * hh) * (Hsum + hh / 2 - fMainForm.yn)
    Tau = fMainForm.F * Abs(Cx1) / fMainForm.D / fMainForm.bi(j)
    Sigma1 = fMainForm.M * fMainForm.Ei(j) * (yy - fMainForm.yn) / fMainForm.D
    Sigma(i) = (Sigma1 ^ 2 + 3 * Tau ^ 2) ^ 0.5
    h(i) = yy
    If Smax < Sigma(i) Then Smax = Sigma(i)
    If Smin > Sigma(i) Then Smin = Sigma(i)
    i = i + 1
    If (zz > sk1 - 1) And (dyy1 <> -1) Then dyy = dyy1
    If zz < (TaskuSkSluoksnyje - 1) Then yy = yy + dyy
    hh = hh + dyy
  Next zz
  Cx = Cx + fMainForm.Ei(j) * fMainForm.Ai(j) * fMainForm.ymi(j)
  Hsum = Hsum + fMainForm.hi(j)
  If Check3.Value = Checked Then
    If Smax < fMainForm.Sui(j) Then Smax = fMainForm.Sui(j)
  End If
Next j
Sigma(i) = 0
h(i) = hmax
Label19.Visible = False
End Select
' Jei reikia apskaiciuoju isvedamo teksto ploti
Dim TekstoPlotis, TekstoAukstis, X00 As Double
If Check2.Value = 1 Then
  txt1 = Picture3.TextWidth(Format(Smax, "0.000") & " MPa")
  txt2 = Picture3.TextWidth(Format(Smin, "0.000") & " MPa")
  If txt1 > txt2 Then TekstoPlotis = txt1 + 2 * Krastai Else TekstoPlotis = txt2 + 2 * Krastai
  TekstoAukstis = Picture3.TextHeight(Format(Smax, "0.000") & " MPa") / 2
Else
  TekstoPlotis = 0
End If
' Braizau profili
Picture3.DrawWidth = 2
Picture3.DrawStyle = vbSolid
Picture3.ForeColor = RGB(0, 0, 0)
If Check1.Value = 1 Then
  Plotis = Picture3.ScaleWidth / 2
  X00 = Plotis - TekstoPlotis
  Picture3.FillStyle = 0
  Mx = (X00 - 2 * Krastai) / bmax
  My = (Picture3.ScaleHeight - 2 * Krastai) / hmax
  If Mx < My Then Mastelis = Mx Else Mastelis = My
  yy = (Picture3.ScaleHeight + hmax * Mastelis) / 2

```

```

dxx = X00 / 2
Picture3.FillStyle = vbUpwardDiagonal
Picture3.DrawWidth = 2
Picture3.ForeColor = RGB(0, 0, 0)
For j = 0 To fMainForm.kiek
    db = Mastelis * fMainForm.bi(j) / 2
    hh = Mastelis * fMainForm.hi(j)
    Picture3.Line (dxx - db, yy)-(dxx + db, yy - hh), , B
    If Picture3.FillStyle = vbUpwardDiagonal Then
        Picture3.FillStyle = vbDownwardDiagonal
    Else
        Picture3.FillStyle = vbUpwardDiagonal
    End If
    yy = yy - hh
Next
Else
    Plotis = Picture3.ScaleWidth - TekstoPlotis
    X00 = 0
    Mastelis = (Picture3.ScaleHeight - 2 * Krastai) / hmax
End If
Y00 = (Picture3.ScaleHeight + hmax * Mastelis) / 2
' Braizau sluoksniu linijas ir surasau maksimalias itempimu reiksmes sluoksniuose
Dim SSmax, X01, X02, X03 As Double
Hsum = 0
If (Check2.Value = 1) And (kiek > 0) Then
    Picture3.DrawWidth = 1
    Picture3.DrawStyle = vbDot
    SSmax = 0
    ha = h(1)
    hs = 0
    X01 = X00 + Plotis + Krastai
    X02 = X00 / 2
    X03 = Picture3.ScaleWidth - Krastai
    Picture3.Line (X02, Y00)-(X03, Y00)
    For i = 1 To kiek - 1
        If Abs(SSmax) < Abs(Sigma(i)) Then SSmax = Sigma(i)
        If Abs(h(i) - h(i + 1)) <= Tikslumas Then
            ha = h(i)
            Picture3.Line (X02, Y00 - Mastelis * ha)-(X03, Y00 - Mastelis * ha)
            Picture3.CurrentX = X01
            Picture3.CurrentY = Y00 - Mastelis * (ha - (ha - hs) / 2) - TekstoAukstis
            Picture3.Print Format(SSmax, "0.000 MPa")
            hs = ha
            SSmax = 0
        End If
    Next i
    Picture3.DrawStyle = vbSolid
End If
If kiek > 0 Then
    ' Apskaiciuoju itempimu masteli
    sMastelis = Abs((Plotis - 2 * Krastai) / (Smax - Smin))
    X00 = X00 + Krastai - sMastelis * Smin
    ' Braizau itempimu epiura
    Picture3.DrawWidth = 2
    Picture3.ForeColor = RGB(0, 0, 128)
    For i = 0 To kiek - 2
        Picture3.Line (X00 + sMastelis * Sigma(i), Y00 - Mastelis * h(i))-(X00 + sMastelis * Sigma(i + 1), Y00
- Mastelis * h(i + 1))
    Next
    Picture3.DrawWidth = 1
    Picture3.ForeColor = RGB(0, 0, 0)
    Picture3.Line (X00, Y00)-(X00, Y00 - Mastelis * hmax)
    ' Jei reikia braizau stiprumo ribas sluoksniuose

```

```

If (Check3.Value = Checked) And (Combo2.ListIndex = 2) Then
    Picture3.DrawWidth = 1
    Picture3.DrawStyle = vbSolid
    Picture3.ForeColor = RGB(196, 0, 0)
    Hsum = 0
    For i = 0 To fMainForm.kiek
        If fMainForm.Sui(i) > 0 Then
            Picture3.Line (X00 + sMastelis * fMainForm.Sui(i), Y00 - Mastelis * Hsum)-(X00 + sMastelis *
fMainForm.Sui(i), Y00 - Mastelis * (Hsum + fMainForm.hi(i)))
            End If
            Hsum = Hsum + fMainForm.hi(i)
        Next
    End If
End If
End Sub

```

```

Public Sub Skerspjuvis(pb As PictureBox, Optional Krastai = 10)
    Dim eile As Integer
    eile = MSHFlexGrid2.RowSel
    If Krastai < 5 Then Krastai = 5
    pb.FillStyle = 0
    pb.Cls
    pb.DrawWidth = 2
    If MSHFlexGrid2.Rows < 3 Then Exit Sub
    hmax = 0
    bmax = 0
    For j = 1 To MSHFlexGrid2.Rows - 1
        MSHFlexGrid2.Row = j
        MSHFlexGrid2.Col = 1
        bh = Val(MSHFlexGrid2.Text)
        If bmax < bh Then bmax = bh
        MSHFlexGrid2.Col = 2
        bh = Val(MSHFlexGrid2.Text)
        hmax = hmax + bh
    Next
    Mx = (pb.ScaleWidth - 2 * Krastai) / bmax
    My = (pb.ScaleHeight - 2 * Krastai) / hmax
    If Mx < My Then Mastelis = Mx Else Mastelis = My
    yy = pb.ScaleHeight - Krastai
    dxx = pb.ScaleWidth / 2
    pb.FillStyle = vbUpwardDiagonal
    Picture3.DrawWidth = 2
    Picture3.ForeColor = RGB(0, 0, 0)
    For j = 1 To MSHFlexGrid2.Rows - 1
        MSHFlexGrid2.Row = j
        MSHFlexGrid2.Col = 1
        db = Mastelis * Val(MSHFlexGrid2.Text) / 2
        MSHFlexGrid2.Col = 2
        hh = Mastelis * Val(MSHFlexGrid2.Text)
        pb.Line (dxx - db, yy)-(dxx + db, yy - hh), , B
        If pb.FillStyle = vbUpwardDiagonal Then pb.FillStyle = vbDownwardDiagonal Else pb.FillStyle =
vbUpwardDiagonal
        yy = yy - hh
    Next
    IsrinktiEilute MSHFlexGrid2, eile
End Sub

```

```

Public Sub Kopijuoti()
    Dim t
    Select Case SSTab1.Tab
        Case 0
            For Each t In Controls
                If TypeOf t Is TextBox Then

```

```

        If t.Tag = 1 Then Clipboard.SetText t SelText
    End If
Next t
Case 1
Clipboard.Clear
If RichTextBox1.SelLength <> 0 Then
    Clipboard.SetText RichTextBox1.SelText
Else
    Clipboard.SetText RichTextBox1.Text
End If
Case 2
Clipboard.Clear
Clipboard.SetData Picture3.Image
End Select
End Sub

Public Function strdel(ByVal eile As String, ByVal nuo As Integer, ByVal kiek As Integer)
    strdel = Mid(eile, 1, nuo) & Mid(eile, nuo + kiek + 1)
End Function

Public Function insertstr(ByVal I_ka As String, ByVal kur As Integer, ByVal ka As String)
    If kur > Len(I_ka) Then
        insertstr = I_ka
    Else
        kaire = Left(I_ka, kur)
        desine = Right(I_ka, Len(I_ka) - kur)
        insertstr = kaire & ka & desine
    End If
End Function

Public Sub Istatyti()
    Dim t
    If SStab1.Tab = 0 Then
        For Each t In Controls
            If TypeOf t Is TextBox Then
                If t.Tag = 1 Then
                    kur = t.SelStart
                    t.Text = strdel(t.Text, kur, Len(t.SelText))
                    t.Text = insertstr(t.Text, kur, Clipboard.GetText)
                    t.SelStart = kur + Len(Clipboard.GetText)
                End If
            End If
        Next t
    End If
End Sub

Public Sub Iskirpti()
    Dim t
    If SStab1.Tab = 0 Then
        For Each t In Controls
            If TypeOf t Is TextBox Then
                If t.Tag = 1 Then
                    Clipboard.SetText t.SelText
                    kur = t.SelStart
                    t.Text = strdel(t.Text, kur, Len(t.SelText))
                    t.SelStart = kur
                End If
            End If
        Next t
    End If
End Sub

```

Formos Apie programa

```
Const KEY_ALL_ACCESS = &H2003F
```

```
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1          ' Unicode nul terminated string
Const REG_DWORD = 4      ' 32-bit number
```

```
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"
```

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long,
ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long)
As Long
```

```
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As
Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As
String, ByRef lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
```

```
Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub
```

```
Private Sub cmdOK_Click()
    Unload Me
End Sub
```

```
Public Sub StartSysInfo()
    On Error GoTo SysInfoErr

    Dim rc As Long
    Dim SysInfoPath As String

    If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath)
Then
        ElseIf GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC,
SysInfoPath) Then
            If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
                SysInfoPath = SysInfoPath & "\MSINFO32.EXE"
            Else
                GoTo SysInfoErr
            End If
        Else
            GoTo SysInfoErr
        End If
        Call Shell(SysInfoPath, vbNormalFocus)
        Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub
```

```
Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef KeyVal
As String) As Boolean
    Dim i As Long          ' Loop Counter
    Dim rc As Long        ' Return Code
```

```

Dim hKey As Long           ' Handle To An Open Registry Key
Dim hDepth As Long       '
Dim KeyValType As Long   ' Data Type Of A Registry Key
Dim tmpVal As String     ' Temporary Storage For A Registry Key Value
Dim KeyValSize As Long   ' Size Of Registry Key Variable
rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open Registry Key
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' Handle Error...
tmpVal = String$(1024, 0) ' Allocate Variable Space
KeyValSize = 1024        ' Mark Variable Size
rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value
If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError ' Handle Errors
tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)
Select Case KeyValType   ' Search Data Types...
Case REG_SZ              ' String Registry Key Data Type
    KeyVal = tmpVal      ' Copy String Value
Case REG_DWORD           ' Double Word Registry Key Data Type
    For i = Len(tmpVal) To 1 Step -1 ' Convert Each Bit
        KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1))) ' Build Value Char. By Char.
    Next
    KeyVal = Format$("&h" + KeyVal) ' Convert Double Word To String
End Select
GetKeyValue = True      ' Return Success
rc = RegCloseKey(hKey) ' Close Registry Key
Exit Function           ' Exit
GetKeyError: ' Cleanup After An Error Has Occured...
KeyValue = ""          ' Set Return Val To Empty String
GetKeyValue = False    ' Return Failure
rc = RegCloseKey(hKey) ' Close Registry Key
End Function

```